Interval temporal logics
ESSLLI 2008 course programme

Valentin Goranko[1] and Angelo Montanari[2]

[1] [1]School of Mathematics, University of the Witwatersrand,
Johannesburg, South Africa
email: goranko@maths.wits.ac.za

[2] [2]Department of Mathematics and Computer Science,
University of Udine, Italy
email: angelo.montanari@dimi.uniud.it

Introduction and general info

This is a tentative outline of the course, which will comprise ten 40-min lectures, delivered 2 per day, with a 5-10 min break in between.

More course material, including updated notes, exercises, slides, and references will be placed on the course web site:

http://www.maths.wits.ac.za/~goranko/esslli08-course.html

Background

Interval-based temporal reasoning arises naturally in a variety of disciplines: philosophy (ontology of time), linguistics (analysis of progressive tenses, semantics and processing of natural languages), artificial intelligence (temporal knowledge representation, systems for time planning and maintenance, theory of events), and computer science (temporal databases, specification, design, and verification of hardware components, concurrent real-time processes, etc.)
Yet, interval temporal logics are far less studied and popular than point-based temporal logics, and one of the main reasons for this is the higher conceptual and computational complexity of the former. Undecidability is a common feature of most systems of interval logic, and this is not surprising, since formulae of these logics translate to *binary relations* over the underlying ordering and, respectively, the validity and satisfiability problems translate into *dyadic second-order logic*. Thus, the search for expressive yet decidable systems of interval logic is a problem of vital importance in that area.
In this course, we will provide a detailed introduction to interval temporal logics and will discuss problems, techniques, and results on expressiveness, axiomatizations, (un)decidability, and tableau-based decision procedures. A number of exercises and open problems will be offered throughout the course.

Prerequisites

Prerequisites: Background in modal and temporal logics and first-order logic. Some background on algorithmic decidability of problems and tableau systems would be useful too.

Course material

This course reader includes, with the kind permission of the publishers, parts of the following publications:

1. V. Goranko, A. Montanari, and G. Sciavicco: A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.
2. V. Goranko, A. Montanari, and G. Sciavicco: Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.
3. 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.
4. D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Complete and terminating tableau for the logic of proper subinterval structures over dense orderings. Proceedings of M4M 4: 5th Workshop on Methods for Modalities, C. Areces and S. Demri (Eds.), November 2007, pp. 335-351 (to appear in the Electronic Notes in Theoretical Computer Science).
5. D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional Interval Neighborhood Logics: Expressiveness, Decidability, and Undecidable Extensions. Research Report 05, Department of Mathematics and Computer Science, University of Udine, Italy, February 2008.

All references have been collected at the end of the reader.

Tentative programme

Day 1: Introduction. Interval structures and relations. Interval logics. Representation theorems.

Lecture 1:

- Brief intro to the course: interval reasoning in AI and CS.
- Interval structures on linear and partial orders. Strict and non-strict interval structures.
- (Binary) relations between intervals. Allen's interval algebra.
- Some important interval structures: neighborhood structures, subinterval structures, etc.
- Ternary relations between intervals: C, D, T.
- Some definabilities between interval relations.

Lecture 2:

- Halpern-Shoham logic HS: syntax and semantics (strict and non-strict). Some definabilities between HS operators.
- Some important fragments of HS: PNL, D-logics, etc.
- Chop-logic. Venema's CDT. Moszkowski's PITL.
- FO characterization of abstract interval structures. Some representation theorems.

Course reading: [50,52]; **additional references:** [2,59]

Day 2: Expressiveness. Undecidability of interval logics.

Lecture 3:

- Standard translation of HS to FOL.
- Comparing expressiveness of interval logics and FOL.
- Some expressive completeness results.
- Bisimulation games. Proving non-definability by using bisimulation games.

Lecture 4:

- Undecidability results for HS and fragments of it.
- Undecidability via tiling.

Course reading: [16,52]; **additional references:** [110]

Day 3: Decidability of interval logics. Tableau-based decision procedures for interval neighborhood logics.

Lecture 5:

- Decidability via semantic restrictions and reduction to point-based logics.
- Decidability of PNL by translation into $FO_2[<]$.
- Other decidability results.

Lecture 6:

- Tableau-based decision procedures for temporal logics: a short account.
- Tableau-based decision procedures for some special cases of interval neighborhood logics.

Course reading: [16,21,52]; **additional references:** [20]

Day 4: Tableau-based decision procedures for the logics of the subinterval relation. General tableau for interval logics.

Lecture 7:

- Tableau-based decision procedures for some special cases of logics of subintervals.

Lecture 8:

- A general tableau system for interval logics.

Course reading: [12,52]; **additional references:** [14]

Day 5: Axiomatic systems for interval logics. Metric interval logics. Conclusions.

Lecture 9:

- Axiomatic systems for interval logics: some results.
- Interval logics which are not finitely axiomatizable.
- Axiomatic systems for interval logics: open problems.

Lecture 10:

- Metric interval logics. Duration Calculus.
- Concluding remarks: research directions and open problems on interval logics.

Course reading: [50,52]; **additional references:** [61,65]

Copy of: **V. Goranko, A. Montanari, and G. Sciavicco: A road map of interval temporal logics and duration calculi. J. of Applied Non-Classical Logics, 14(1-2):9–54, 2004**
Included with permission from Hermes Publishers.

Abstract. We survey main developments, results, and open problems on interval temporal logics and duration calculi. We present various formal systems studied in the literature and discuss their distinctive features, emphasizing on expressiveness, axiomatic systems, and (un)decidability results.

1 Introduction

Interval-based temporal logics stem from four major scientific areas:

Philosophy. The philosophical roots of interval temporal logics can be traced back to Zeno and Aristotle. The nature of Time has always been a favourite subject in philosophy, and in particular, the discussion whether time instants or time periods should be regarded as the primary objects of temporal ontology has a distinct philosophical flavour. Some of the modern formal logical treatments of interval-based structure of time include: [60] providing a philosophical analysis of interval ontology and interval based tense logics; [67] which elaborates on Hamblin's work, and introduces a sequent calculus for an interval tense logic over precedence and sub-interval relations; [102], a follow-up on Humberstone's work, discussing and analyzing persistency (preservation of truth in sub-intervals) and homogeneity; [22] proposing axiomatic systems for interval-based tense logics of the rationals and reals, studied earlier in [102]. A comprehensive study and logical analysis of point-based and interval-based ontologies, languages, and logical systems can be found in [7].

Linguistics. Interval-based logical formalisms have featured in the study of natural languages since the seminal work of Reichenbach [100]. They arise as suitable frameworks for modeling progressive tenses and expressing various language constructions involving time periods and event duration which cannot be adequately grasped by point-based temporal languages. Period-based temporal languages and logics have been proposed and studied e.g. in [35,70,101]. The linguistic aspects of interval logics will not be treated here, apart from some discussion of the expressiveness of various interval-based temporal languages.

Artificial intelligence. Interval temporal languages and logics have sprung up from *expert systems*, *planning systems*, *temporal databases*, *theory of events*, *natural language analysis and processing*, etc. as formal tools for temporal representation and reasoning in artificial intelligence. Some of the notable contributions in that area include: [2] proposing a temporal logic for reasoning of time intervals and the 13 Allen's relations between intervals in a linear ordering; [4] providing an axiomatization and representation result of interval structures based on the *meet* relation between intervals, further advanced and studied in [73] which also presents a completeness theorem and algorithms for satisfiability checking for Allen's calculus represented as a first-order theory; [46] critically analyzing Allen's framework and arguing the necessity of considering points and intervals on a par, and [3] developing interval-based theory of actions and events. For a recent survey on temporal representation and reasoning in artificial intelligence see [27].

Computer science. One of the first applications of interval temporal logics to computer science, viz. for specification and design of hardware components, was proposed in [58] and [84], further developed in [90,86,87,88]. Later, other systems and applications of interval logics were proposed in [29,30,33,34,9,114,98], and model checking tools and techniques for interval logics were developed and applied in [23,94]. Particularly suitable interval logics for specification and verification of real-time processes in computer science are the *duration calculi* (see [116,107,63,117,62,113]) introduced as extensions of interval logics, allowing representation and reasoning about time durations for which a system is in a given state. For an up-to-date survey on duration calculi see [61].

In this survey we outline (without claiming completeness) main developments, results, and open problems on interval temporal logics and duration calculi. We present various formal systems studied in the literature and discuss their distinctive features, emphasizing on expressiveness, axiomatic systems, and (un)decidability results. Since duration calculi are discussed in more details in [61], we will survey this topic rather briefly, while going in more detail on interval logics, mainly on propositional level.
The survey is organized as follows. In Section 2, we introduce the basic syntactic and semantic ingredients of interval temporal logics and duration calculi, including interval temporal structures, operators, and languages with their syntax and semantics. In Section 3 we present propositional interval logics, and in Section 4 first-order interval logics and duration calculi. Section 5 contains a brief summary of other relevant results not presented in the survey, while the Appendix contains some proofs of important results mentioned in the main text.

2 Preliminaries

2.1 Temporal ontologies, interval structures and relations between intervals

Interval temporal logics are subject to the same ontological dilemmas as the instant-based temporal logic, viz.: should the time structure be considered *linear* or *branching? Discrete* or *dense? With* or *without beginning?* etc. In addition, however, new dilemmas arise regarding the nature of the intervals:

- *Should intervals include their end-points or not?*
- *Can they be unbounded?*
- *Are point-intervals (i.e. with coinciding endpoints) admissible or not?*
- *How are points and intervals related? Which is the primary concept? Should an interval be identified with the set of points in it, or there is more into it?*

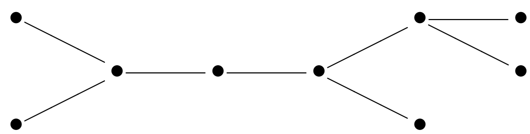
The last question is of particular importance when semantics of interval logics are defined.

Given a strict partial ordering $\mathbb{D} = \langle D, < \rangle$ an **interval** in $\mathbb{D}$ is a pair $[d_0, d_1]$ such that $d_0, d_1 \in \mathbb{D}$ and $d_0 \leq d_1$. $[d_0, d_1]$ is a **strict** interval if $d_0 < d_1$. Often we will refer to all intervals on $\mathbb{D}$ as **non-strict intervals**, to distinguish from the latter. In particular, intervals $[d, d]$ will be called **point-intervals**. A point d *belongs to an interval* $[d_0, d_1]$ if $d_0 \leq d \leq d_1$ (i.e. the endpoints of an intervals are included in it). The set of all non-strict intervals on $\mathbb{D}$ will be denoted by $\mathbb{I}^+(\mathbb{D})$, while the set of all strict intervals will be denoted by $\mathbb{I}^-(\mathbb{D})$. By $\mathbb{I}(\mathbb{D})$ we will denote either of these. For the purpose of this survey, we will call a pair $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ an **interval structure**.

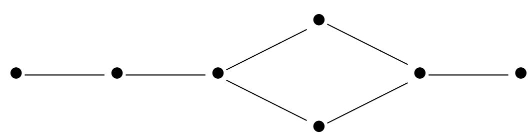
In all systems considered here the intervals will be assumed *linear*, although this restriction can often be relaxed without essential complications. Thus, we will concentrate on partial orderings with the **linear intervals property:**

$$\forall x \forall y (x < y \rightarrow \forall z_1 \forall z_2 (x < z_1 < y \wedge x < z_2 < y \rightarrow z_1 < z_2 \vee z_1 = z_2 \vee z_2 < z_1)),$$

that is, orderings in which every interval is linear. Clearly every linear ordering falls here. An example of a non-linear ordering with this property is:



while a non-example is:



An interval structure is:

- **linear** if every two points are comparable;
- **discrete,** if every point with a successor/predecessor has an immediate successor/predecessor along every path starting from/ending in it, that is,

  $$\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z \leq y \wedge \forall w (x < w \wedge w \leq y \rightarrow z \leq w))),$$

  and

  $$\forall x \forall y (x < y \rightarrow \exists z (x \leq z \wedge z < y \wedge \forall w (x \leq w \wedge w < y \rightarrow w \leq z)));$$

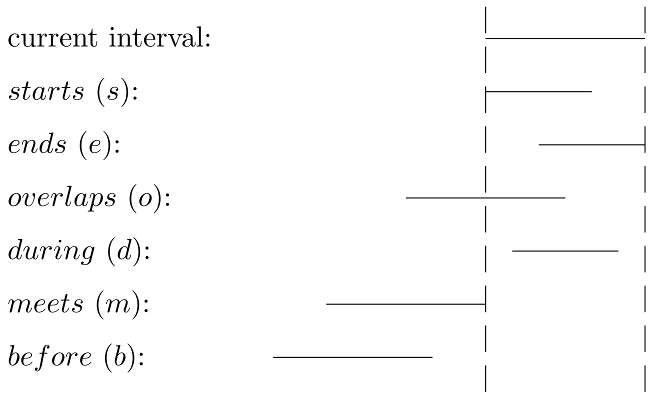
- **dense,** if for every pair of different comparable points there exists another point in between:

  $$\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y));$$

- **unbounded above** (resp. **below**) if every point has a successor (resp. predecessor);
- **Dedekind complete** if every non-empty and bounded above set of points has a least upper bound.

Besides interval logics over the classes of linear, (un)bounded, discrete, dense, and Dedekind complete interval structures, we will be discussing those interpreted on the single structures $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$, and $\mathbb{R}$ with their usual orderings.

It is well known that there are 13 different binary relations between intervals on a linear ordering (and quite a few more on a partial ordering) [2]:

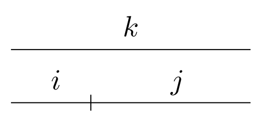


These relations lead to a rich interval algebra, which will not be discussed in detail here. (A survey of Allen's interval algebra and of a number of its tractable fragments, including Vilain and Kautz's Point Algebra [111], van Beek's Continuous Endpoint Algebra [108], and Nebel and Bürckert's ORD-Horn Algebra [91], can be found in [27].)

Another natural binary relation between intervals, definable in terms of Allen's relations, is the one of *sub-interval* which comes in three versions. Given a partial ordering $\langle D, < \rangle$ and intervals $[s_0, s_1]$ and $[d_0, d_1]$ in it:

- $[s_0, s_1]$ is a **sub-interval** of $[d_0, d_1]$ if $d_0 \leq s_0$ and $s_1 \leq d_1$. The relation of sub-interval will be denoted by $\sqsubseteq$;
- $[s_0, s_1]$ is a **proper sub-interval** of $[d_0, d_1]$, denoted $[s_0, s_1] \sqsubset [d_0, d_1]$, if $[s_0, s_1] \sqsubseteq [d_0, d_1]$ and $[s_0, s_1] \neq [d_0, d_1]$;
- $[s_0, s_1]$ is a **strict sub-interval** of $[d_0, d_1]$, denoted $[s_0, s_1] \prec [d_0, d_1]$, if $d_0 < s_0$ and $s_1 < d_1$.

Amongst the multitude of *ternary* relations between intervals there is one of particular importance for us, which correspond to the binary operation of concatenation of meeting intervals. Such a ternary interval relation A, which has been introduced by Venema in [110], can be graphically depicted as follows:



It is defined as follows:

$$Aijk \;\text{ if }\; i \textit{ meets } j,\ i \textit{ begins } k,\ \textit{ and } j \textit{ ends } k,$$

that is, k is the concatenation of i and j.

2.2 Propositional interval temporal languages and models

The generic language of propositional interval logics includes the set of propositional letters $\mathcal{AP}$, the classical propositional connectives $\neg$ and $\wedge$ (all others, including the propositional constants $\top$ and $\bot$, are definable as usual), and a set of *interval temporal operators (modalities)* specific for each logical system.

There are two different natural semantics for interval logics, namely, a **strict** one, which excludes point-intervals, and a **non-strict** one, which includes them. A **non-strict interval model** is a structure $\mathbf{M}^{+}= \langle \mathbb{D}, V\rangle$, where $\mathbb{D}$ is a partial ordering and $V : \mathbb{I}^{+}(\mathbb{D}) \rightarrow \mathbf{P}(\mathcal{AP})$ is a **valuation** assigning to each interval a set of atomic propositions considered true at it. Respectively, a **strict interval model** is a structure $\mathbf{M}^{-}= \langle \mathbb{D}, V\rangle$ defined likewise, where $V : \mathbb{I}^{-}(\mathbb{D}) \rightarrow \mathbf{P}(\mathcal{AP})$. When we do not wish to specify the strictness, we will write simply $\mathbf{M}$, assuming either version.

Allen's relations give rise to respective unary modal operators, thus defining the modal logic of time intervals HS introduced by Halpern and Shoham in [59]. Some of these modal operators are definable in terms of others and it suffices to choose as basic the modalities corresponding to the relations **begin, end**, and their inverses. Thus, the formulas of HS are generated by the following abstract syntax:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle B\rangle\phi \mid \langle E\rangle\phi \mid \langle\overline{B}\rangle\phi \mid \langle\overline{E}\rangle\phi.$$

The formal semantics of these modal operators (given in [59] in terms of non-strict models) is defined as follows:

($\langle B\rangle$) $\mathbf{M}^{+}, [d_0, d_1] \Vdash \langle B\rangle\phi$ if there exists d_2 such that $d_0 \leq d_2 < d_1$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.
($\langle E\rangle$) $\mathbf{M}^{+}, [d_0, d_1] \Vdash \langle E\rangle\phi$ if there exists d_2 such that $d_0 < d_2 \leq d_1$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.
($\langle\overline{B}\rangle$) $\mathbf{M}^{+}, [d_0, d_1] \Vdash \langle\overline{B}\rangle\phi$ if there exists d_2 such that $d_1 < d_2$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.
($\langle\overline{E}\rangle$) $\mathbf{M}^{+}, [d_0, d_1] \Vdash \langle\overline{E}\rangle\phi$ if there exists d_2 such that $d_2 < d_0$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.

A useful new symbol is the *modal constant* π *for point-intervals* interpreted in non-strict models as follows:

$$\mathbf{M}, [d_0, d_1] \Vdash \pi \text{ if } d_0 = d_1.$$

Note that the constant π is definable as either $[B]\bot$ or $[E]\bot$, so it is only needed in weaker languages. The presence of π in the language allows for interpretation of the strict semantics into the non-strict one, by means of the translation:

- $\tau(p) = p$ for $p \in \mathcal{AP}$;
- $\tau(\neg\varphi) = \neg\tau(\varphi)$;
- $\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$;
- $\tau(\langle * \rangle \varphi) = \langle * \rangle (\neg\pi \wedge \tau(\varphi))$ for any (unary) interval diamond-modality $\langle * \rangle$.

The interpretation is effected by the following claim, proved by a straightforward induction on φ :

Proposition 1. *For every interval model* $\mathbf{M}$, *proper interval* $[d_0, d_1]$ *in* $\mathbf{M}$, *and a formula* φ, $\mathbf{M}^{-}, [d_0, d_1] \Vdash \varphi$ *iff* $\mathbf{M}^{+}, [d_0, d_1] \Vdash \tau(\varphi)$.

Usually, but not always, the non-strict semantics is taken by default.

Accordingly, the (non-strict) semantics of the binary modalities C, D, and T, associated with the ternary relation A, is:

(C) $\mathbf{M}^{+}, k \Vdash \phi C\psi$ iff there exists intervals i, j such that $Aijk$ and $\mathbf{M}^{+}, i \Vdash \phi$, and $\mathbf{M}^{+}, j \Vdash \psi$, that is:
$\mathbf{M}^{+}, [d_0, d_1] \Vdash \phi C\psi$ iff there exists $d_2 \in \mathbb{D}$ such that $d_0 \leq d_2 \leq d_1$, $\mathbf{M}^{+}, [d_0, d_2] \Vdash \phi$, and $\mathbf{M}^{+}, [d_2, d_1] \Vdash \psi$.

(D) $\mathbf{M}^+, j \Vdash \phi D \psi$ iff there exists intervals i, k such that $Aijk$ and $\mathbf{M}^+, i \Vdash \phi$, and $\mathbf{M}^+, k \Vdash \psi$, that is:
$\mathbf{M}^+, [d_0, d_1] \Vdash \phi D \psi$ iff there exists $d_2 \in \mathbb{D}$ such that $d_2 \leq d_0$, $\mathbf{M}^+, [d_2, d_0] \Vdash \phi$, and $\mathbf{M}^+, [d_2, d_1] \Vdash \psi$.
(T) $\mathbf{M}^+, i \Vdash \phi T \psi$ iff there exists intervals j, k such that $Aijk$ and $\mathbf{M}^+, j \Vdash \phi$, and $\mathbf{M}^+, k \Vdash \psi$, that is:
$\mathbf{M}^+, [d_0, d_1] \Vdash \phi T \psi$ iff there exists $d_2 \in \mathbb{D}$ such that $d_1 \leq d_2$, $\mathbf{M}^+, [d_1, d_2] \Vdash \phi$, and $\mathbf{M}^+, [d_0, d_2] \Vdash \psi$.

The semantics of interval temporal logics is sometimes subjected to restrictions justified for specific applications for which a logical system is designed, such as:

- **locality**, meaning that all atomic propositions are point-wise and truth at an interval is defined as truth at its initial point.
- **homogeneity**, requiring that truth of a formula at an interval implies truth of that formula at every sub-interval.

Also, in the so called **split-structures** not all sub-intervals of an interval are 'available' but only those two which are determined by the 'split-point' in that interval.
We will not assume any semantic restrictions, unless otherwise specified.

2.3 First-order languages and models for interval logics and duration calculi

The first-order languages for interval logics extend the propositional ones essentially the same way as in classical logic, but accounting for the fact that the first-order domain may change over time. Formally, these languages involve **terms** built as usual from variables, constants and functional symbols. Constants and functional symbols are classified as *global* (or *rigid*) (whose interpretation does not depend on the time) and *temporal* (or *flexible*) (whose interpretation can vary over time). Predicate symbols (also classified as global or temporal) are denoted by $p^i, q^j, \ldots$, where $i, j \ldots$ represent the arities. The abstract syntax of formulas of a generic first-order interval language includes the clauses

$$\phi ::= p^n(\theta_1, \ldots, \theta_n) \mid \exists x \phi \mid \ \neg \phi \mid \phi \wedge \psi$$

where $\theta_1, \ldots, \theta_n$ are terms, plus the clauses for the specific interval modal operators.
Among the constants, there is a specific and important one, present in most of the first-order languages for interval logics and duration calculi, viz. the flexible constant l denoting the **length of the current interval**. Often it is combined with a structure of an additive group (typically, the additive group of reals) as part of the temporal domain, which allows for computing lengths of concatenated intervals, etc.
A specific additional feature of the syntax of duration calculi is the special category of terms called **state expressions** which are meant to represent the duration for which a system stays in a particular state.
The semantics of first-order interval formulas is a combination of the standard semantics of a first-order (temporal) logic with the semantics of the specific underlying propositional interval logic.

3 Propositional Interval Logics

As already noted, every interval logic L has two versions: the *strict* L^- and the *non-strict* L^+, and when writing just L we will mean the non-strict one.

3.1 Monadic interval logics

Here we will introduce and discuss briefly the most well-known (or just interesting) interval logics involving only unary modal operators, starting from the weakest.

3.1.1 The sub-interval logic D. Perhaps the most natural relations between intervals are those of *sub-interval* and *meet*. The latter corresponds to the neighborhood logics which will be discussed later. We denote the generic interval logic based on the former by D. The abstract syntax of the simplest version of D is:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle D\rangle\phi,$$

but one could also include in the language the modal constant π.

The sub-interval relation and the temporal logics associated with it were studied, from the perspective of philosophical temporal logics, in [60,102], [67] (together with precedence), and [7]. In the computer science literature, it was apparently first mentioned in [59] and its expressiveness (interpreted over linear non-strict models) discussed in [74].

Besides the strict and non-strict versions, the logic D allow essential semantic variations, depending on which sub-interval relation ($\sqsubseteq$, $\sqsubset$, or $\prec$) is assumed. Accordingly, the truth definition for D is based on the clause:

($\langle D\rangle$) $\mathbf{M}, [d_0, d_1] \Vdash \langle D\rangle\phi$ iff there exists a sub-interval $[d_2, d_2]$ of $[d_0, d_1]$ such that $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

At present, we are not aware of any specific published results about expressive power, axiomatic systems, and decidability for (variants of) the logic D, but we note that, at least in the cases of proper and strict versions, non-trivial validities expressible in D arise, associated with length vs depth (maximal length of chains of nested sub-intervals).

3.1.2 The logic BE. The logic BE features the two modalities $\langle B\rangle$ and $\langle E\rangle$, and its formulas are generated by the following abstract syntax:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle B\rangle\phi \mid \langle E\rangle\phi.$$

As we already shown, the modal constant π is definable as $[B]\bot$. Accordingly, the point-intervals that respectively begins and ends the current interval can be captured as follows:

- $[[BP]]\phi \triangleq (\phi \wedge \pi) \vee \langle B\rangle(\phi \wedge \pi)$, and
- $[[EP]]\phi \triangleq (\phi \wedge \pi) \vee \langle E\rangle(\phi \wedge \pi)$.

The logic BE is at least as expressive as D. Indeed, the modality $\langle D\rangle$ can be defined as

- $\langle D\rangle\phi \triangleq \langle B\rangle\langle E\rangle\phi$.

The undefinability of $\langle B\rangle$ and $\langle E\rangle$ in D has been conjectured by Lodaya in [74], but no formal proof was given.
BE is expressive enough to capture some relevant conditions on the underlying interval structure (as originally pointed out by Halpern and Shoham in the context of the logic HS [59]). First, one can constrain an interval structure to be discrete by means of the formula

- **discrete** $\triangleq \pi \vee l1 \vee (\langle B\rangle l1 \wedge \langle E\rangle l1)$,

where $l1$ is true over an interval $[d_0, d_1]$ if and only if $d_0 < d_1$ and there are no points between d_0 and d_1. Such a condition can be expressed in BE as follows:

$l1 \triangleq \langle B\rangle\top \wedge [B][B]\bot$.

It is not difficult to show that an interval structure is discrete if and only if the formula **discrete** is valid in it. Furthermore, one can easily force an interval structure to be dense by constraining the formula

- **dense** $\triangleq \neg l1$.

to be valid. Finally, one can constrain an interval structure to be Dedekind complete by means of the formula

- **Dedekind complete** $\triangleq (\langle B\rangle cell \wedge [[EP]]\neg q \wedge [E]([[BP]]q \rightarrow \langle B\rangle cell))$
  $\rightarrow \langle B\rangle([E](\neg\pi \rightarrow \langle D\rangle cell))$

where *cell* is true over an interval $[d_0, d_1]$ if and only if its begin and end points satisfy a given proposition letter q (the cell delimiters), all sub-intervals satisfy a proposition letter p (the cell content), and there exists at least one sub-interval satisfying p, that is,

$cell \triangleq [[BP]]q \wedge [[EP]]q \wedge [D]p \wedge \langle D\rangle p$.

BE also allows one to define the *universal modality* $[All]$ (the application of $[All]$ to a formula φ constrains φ to hold over every interval of the model), which is captured by the following formula:

- $[All]\phi \triangleq [B]\phi \wedge [E]\phi \wedge [B][E]\phi$.

As for (un)decidability results, Lodaya [74] proves the following theorem:

Theorem 1. *The satisfiability problem for* BE*-formulas interpreted over non-strict dense linear structures is not decidable.*

The structure of the proof is outlined in the Appendix. As mentioned earlier, we do not known whether the satisfiability problem for D over dense structures is decidable. Because density is expressible in BE by a constant formula and the universal modality is definable in the same logic, it follows that:

Corollary 1. *The satisfiability problem for* BE *over the class of all non-strict linear structures is not decidable.*

Indeed, the satisfiability of a formula ϕ in a dense model is equivalent to the satisfiability of $[All]\neg l1 \wedge \phi$ in any non-strict model.
We conclude our description of BE by pointing out that a number of meaningful problems, such as the decidability of the satisfiability problem for BE-formulas interpreted over special classes of linear ordering, or over strict models, and the definition of sound and complete axiomatic systems for BE, are, at the best of our knowledge, still open.

3.1.3 Propositional neighborhood logics. The interval logics based on the *meet* relation and its inverse *met-by* are called **neighborhood logics**. Notably, first-order neighborhood logics were introduced and studied by Zhou and Hansen in [114] before their propositional variants were studied only quite recently over linear structures (both strict and non-strict) by Goranko, Montanari, and Sciavicco [50].

The language of propositional neighborhood logics includes the modal operators $\Diamond_r$ and $\Diamond_l$ borrowed from [114]. Its formulas are generated by the following abstract syntax:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \Diamond_r\phi \mid \Diamond_l\phi.$$

The dual operators $\Box_r$ and $\Box_l$ are defined in the usual way. To make it easier to distinguish between the two semantics from the syntax, we will reserve this notation for the case of non-strict propositional neighborhood logics, generically denoted by PNL^+, while for the strict ones, denoted by PNL^-, $\langle A\rangle$ and $\langle\overline{A}\rangle$ are used instead of $\Diamond_r$ and $\Diamond_l$, respectively. The class of non-strict propositional neighborhood logics extended with the modal constant π will be denoted by $\text{PNL}^{\pi+}$.
It is worth noticing that $\langle A\rangle$ and $\langle\overline{A}\rangle$ were originally introduced in the logic HS [59] as derived operators. The semantics of HS admits point-intervals and hence, according to our classification, it is non-strict. However, the modalities $\langle A\rangle$ and $\langle\overline{A}\rangle$ only refer to strict intervals, and thus the semantics of the fragment $\langle A\rangle\langle\overline{A}\rangle$ can be considered essentially strict.

The formal semantics of the modal operators $\Diamond_r$ and $\Diamond_l$ is defined as follows:

($\Diamond_r$) $\mathbf{M}^+, [d_0, d_1] \Vdash \Diamond_r\phi$ if there exists d_2 such that $d_1 \leq d_2$ and $\mathbf{M}^+, [d_1, d_2] \Vdash \phi$;
($\Diamond_l$) $\mathbf{M}^+, [d_0, d_1] \Vdash \Diamond_l\phi$ iff there exists d_2 such that $d_2 \leq d_0$ and $\mathbf{M}^+, [d_2, d_0] \Vdash \phi$,

while the semantic clauses for the operators $\langle A\rangle$ and $\langle\overline{A}\rangle$ are:

($\langle A\rangle$) $\mathbf{M}^-, [d_0, d_1] \Vdash \langle A\rangle\phi$ if there exists d_2 such that $d_1 < d_2$ and $\mathbf{M}^-, [d_1, d_2] \Vdash \phi$;
($\langle\overline{A}\rangle$) $\mathbf{M}^-, [d_0, d_1] \Vdash \langle\overline{A}\rangle\phi$ iff there exists d_2 such that $d_2 < d_0$ and $\mathbf{M}^-, [d_2, d_0] \Vdash \phi$.

Propositional neighborhood logics are quite expressive. For example, in the strict semantics we can characterize various classes of linear structures:

(A-SPNL^u) $[A]p \rightarrow \langle A\rangle p$, in conjunction with its inverse, defines the class of **unbounded** structures;
(A-SPNL^d) $(\langle A\rangle\langle A\rangle p \rightarrow \langle A\rangle\langle A\rangle\langle A\rangle p) \wedge (\langle A\rangle[A]p \rightarrow \langle A\rangle\langle A\rangle[A]p)$, in conjunction with its inverse, defines the class of **dense** structures, extended with the 2-element linear ordering[3];
(A-SPNL^z) $\langle A\rangle[\overline{A}](p \wedge [\overline{A}]p) \rightarrow \langle A\rangle[A][\overline{A}][\overline{A}](p \wedge [\overline{A}]p)$, in conjunction with its inverse, defines the class of **discrete** structures;
(A-SPNL^c) $\langle A\rangle\langle A\rangle[\overline{A}]p \wedge \langle A\rangle[A]\neg[\overline{A}]p \rightarrow \langle A\rangle(\langle A\rangle[\overline{A}]\ [\overline{A}]p \wedge\ [A]\ \langle A\rangle\neg\ [\overline{A}]\ p)$ defines the class of **Dedekind complete** structures.

Moreover, the language of PNL^- over unbounded structures is powerful enough to express the **difference** $[\neq]$ operator:

$$[\neq](q) \equiv [\overline{A}][\overline{A}][A]q \wedge [\overline{A}][A][A]q \wedge [A][A][\overline{A}]q \wedge [A][\overline{A}][\overline{A}]q,$$

[3] The 2-element linear ordering cannot be separated in the language of PNL^-.

saying that q is true at some other interval, and consequently to simulate **nominals**: $(n(q) \equiv q \wedge [\neq](\neg q))$ expresses the claim that q holds in the current interval and nowhere else. It follows (see, e.g., [47]) that every universal property of strict unbounded linear models can be expressed in PNL$^-$.

Sound and complete axiomatic systems for propositional neighborhood logics are given in [50].

Theorem 2. *The following axiomatic system is sound and complete for the logic* PNL$^+$ *of non-strict linear models:*

(A-NSPNL1) enough propositional tautologies;
(A-NSPNL2) the K axioms for $\Box_r$ and $\Box_l$;
(A-NSPNL3) $\Box_r p \rightarrow \Diamond_r p$, and its inverse;
(A-NSPNL4) $p \rightarrow \Box_r \Diamond_l p$, and its inverse;
(A-NSPNL5) $\Diamond_r \Diamond_l p \rightarrow \Box_r \Diamond_l p$, and its inverse;
(A-NSPNL6) $\Box_r \Diamond_l p \rightarrow \Diamond_l \Diamond_r \Diamond_r p \vee \Diamond_l \Diamond_l \Diamond_r p$, and its inverse;
(A-NSPNL7) $\Diamond_r \Diamond_r \Diamond_r p \rightarrow \Diamond_r \Diamond_r p$, and its inverse;
(A-NSPNL8) $\Box_r q \wedge \Diamond_r p_1 \wedge \ldots \wedge \Diamond_r p_n \rightarrow \Diamond_r(\Box_r q \wedge \Diamond_r p_1 \wedge \ldots \wedge \Diamond_r p_n)$, and its inverse, for each $n \geq 1$.

Rules of inference: Modus Ponens, Uniform Substitution, and $\Box_r$ and $\Box_l$-generalization. Interestingly, some of these axioms, including the infinite scheme (A-NSPNL8), were not included in the axiomatization of the first-order neighborhood logic given in [5] as they could be derived using the first-order axioms.

Theorem 3. *A sound and complete axiomatic system for the logic* PNL$^{\pi+}$ *can be obtained from that for* PNL$^+$ *by adding the following axioms:*

(A-NSPNL$^\pi$1) $\Diamond_l \pi \wedge \Diamond_r \pi$*;*
(A-NSPNL$^\pi$2) $\Diamond_r(\pi \wedge p) \rightarrow \Box_r(\pi \rightarrow p)$ *and its inverse;*
(A-NSPNL$^\pi$3) $\Diamond_r p \wedge \Box_r q \rightarrow \Diamond_r(\pi \wedge \Diamond_r p \wedge \Box_r q)$ *and its inverse.*

Once $\Diamond_r, \Diamond_l$ are substituted by $\langle A \rangle, \langle \overline{A} \rangle$, and $\Box_r, \Box_l$ accordingly by $[A], [\overline{A}]$, the axioms for PNL$^-$ are very similar to those for PNL$^+$ (accordingly modified to reflect the fact that point-intervals are now excluded), except for the scheme (A-NF$_\infty$) which is no longer valid.

Theorem 4. *The following axiomatic system is sound and complete for the logic* PNL$^-$ *of strict linear models:*

(A-SPNL1) *enough propositional tautologies;*
(A-SPNL2) *the K axioms for $[A]$ and $[\overline{A}]$;*
(A-SPNL3) $p \rightarrow [A]\langle \overline{A} \rangle p$ *and its inverse.*
(A-SPNL4) $\langle A \rangle \langle \overline{A} \rangle p \rightarrow [A]\langle \overline{A} \rangle p$ *and its inverse;*
(A-SPNL5) $(\langle \overline{A} \rangle \langle \overline{A} \rangle \top \wedge \langle A \rangle \langle \overline{A} \rangle p) \rightarrow p \vee \langle \overline{A} \rangle \langle A \rangle \langle A \rangle p \vee \langle \overline{A} \rangle \langle \overline{A} \rangle \langle A \rangle p$ *and its inverse;*
(A-SPNL6) $\langle A \rangle \langle A \rangle \langle A \rangle p \rightarrow \langle A \rangle \langle A \rangle p$ *and its inverse.*

Let us denote the strict neighborhood logic respectively interpreted over unbounded, dense, discrete, Dedekind complete, unbounded and dense, unbounded and discrete, and unbounded and Dedekind complete linear structures by PNL$^{\lambda-}$, where $\lambda \in \{u, d, z, c, ud, uz, uc\}$ respectively. Likewise, PNL$^{\lambda+}$ denotes the respective classes of non-strict models.

Theorem 5. *The following results hold:*

1. *For every* $\lambda_1, \lambda_2 \in \{u, d, z, c, ud, uz, uc\}$*,* $\mathrm{PNL}^{\lambda_1 -} \subsetneq \mathrm{PNL}^{\lambda_2 -}$ *if and only if the class of linear orders characterized by the condition* λ_2 *is strictly contained in the class of linear orders characterized by the condition* λ_1*.*
2. $\mathrm{PNL}^{ud-} \subsetneq \mathrm{PNL}^{+}$*, where the inclusion is in terms of the obvious translation between the two languages.*
3. $\mathrm{PNL}^{+} = \mathrm{PNL}^{u+} = \mathrm{PNL}^{d+} = \mathrm{PNL}^{ud+} = \mathrm{PNL}^{z+} = \mathrm{PNL}^{uz+}$.

We also note that the logic PNL^{uz-} does not yet characterize the interval structure of the integers, because the formula

$$\langle A\rangle p \wedge [A](p \rightarrow \langle A\rangle p) \wedge [A][A](p \rightarrow \langle A\rangle p) \rightarrow [A]\langle A\rangle\langle A\rangle p$$

is valid in the integers, but not in PNL^{uz-} since it fails in a PNL^{uz-}-model based on $\mathbb{Z} + \mathbb{Z}$.

Theorem 6. *The following completeness results hold:*

1. *The axiomatic system for* PNL^{-} *extended with* A-SPNL^{u} *is sound and complete for the class of unbounded structure.*
2. *The axiomatic system for* PNL^{-} *extended with* A-SPNL^{d} *is sound and complete for the class of dense structures.*
3. *The axiomatic system for* PNL^{-} *extended with* A-SPNL^{z} *is sound and complete for discrete structures.*
4. *The axiomatic system for* PNL^{-} *extended with* A-SPNL^{ud} *is sound and complete for the class of dense unbounded structures.*
5. *The axiomatic system for* PNL^{-} *extended with* A-SPNL^{uz} *is sound and complete for the class of discrete unbounded structures.*

Finally, we note that most of the decidability problems related to propositional neighborhood logics are still open.

3.1.4 The logic HS. The most expressive propositional interval logic with unary modal operators studied in the literature is Halpern and Shoham's logic HS introduced in [59]. HS contains (as primitive or definable) all unary modalities introduced earlier. As anticipated in Section 2, HS features the modalities $\langle B\rangle, \langle E\rangle$ and their inverses $\langle \overline{B}\rangle, \langle \overline{E}\rangle$, which suffice to define all other modal operators, so that it can be regarded as the temporal logic of Allen's relations. Unlike most other previously studied interval logics, HS was originally interpreted in non-strict models not over linear orderings, but over all partial orderings with the linear intervals property.

Formally, HS-formulas are generated by the following abstract syntax:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle B\rangle\phi \mid \langle E\rangle\phi \mid \langle \overline{B}\rangle\phi \mid \langle \overline{E}\rangle\phi.$$

Furthermore, as pointed out by Venema in [109], the neighborhood modalities $\langle A\rangle$ and $\langle \overline{A}\rangle$ are definable in the non-strict semantics as follows:

- $\langle A\rangle\phi \triangleq [[EP]]\overline{B}\phi$, and
- $\langle \overline{A}\rangle\phi \triangleq [[BP]]\overline{E}\phi$.

HS can express linearity of the interval structure by means of the following formula:

- **linear** $\triangleq (\langle A\rangle p \rightarrow [A](p \vee \langle B\rangle p \vee \langle \overline{B}\rangle p)) \wedge (\langle \overline{A}\rangle p \rightarrow [\langle \overline{A}\rangle](p \vee \langle E\rangle p \vee \langle \overline{E}\rangle p))$,

as well as all conditions that can be expressed in its fragment BE.

As expected, HS is a highly undecidable logic. In [59] the authors have obtained important results about non-axiomatizability, undecidability and complexity of the satisfiability in HS for many natural classes of models. Their idea for proving undecidability is based on using an infinitely ascending sequence in the model to simulate the halting problem for Turing Machines. An **infinitely ascending sequence** is an infinite sequence of points $d_0, d_1, d_2, \ldots$ such that $d_i < d_{i+1}$ for all i. Any unbounded above ordering contains an infinite ascending sequence. A class of ordered structures contains an infinite ascending sequence if at least one of the structures in the class does.

Theorem 7. *The validity problem in* HS *interpreted over any class of ordered structures with an infinitely ascending sequence is r.e.-hard.*

Thus, in particular, HS is undecidable for the class of all (non-strict) models, the class of all linear models, the class of all discrete linear models, the class of all dense linear models, and the class of all dense and unbounded linear models. An outline of the proof of the above theorem can be found in the Appendix.

Theorem 8. *The validity problem in* HS *interpreted over any class of Dedekind complete ordered structures having an infinitely ascending sequence is* Π^1_1*-hard.*

For instance, the validity in HS in any of the orderings of the natural numbers, integers, or reals is not recursively axiomatizable. Undecidability occurs even without existence of infinitely ascending sequences. A class of ordered structures has **unboundedly ascending sequences** is for every n there is a structure in the class with an ascending sequence of length at least n.

Theorem 9. *The validity problem in* HS *interpreted over any class of Dedekind complete ordered structures having unboundedly ascending sequences is co-r.e. hard.*

Another proof of undecidability of HS, using a tiling problem, is given in [45].
In [109] (see also [77]) Venema has shown that HS interpreted over a linear ordering is at least as expressive as the universal monadic second-order logic, where second-order quantification is only allowed over monadic predicates, and there are cases where it is strictly more expressive. As a corollary, it can be proved that HS is strictly more expressive than every point-based temporal logic on linear orderings.

In the same paper Venema provided an interesting *geometrical* interpretation of HS, using which he obtained sound and complete axiomatic systems for HS with respect to relevant classes of structures. Here is the idea. An interval can be viewed as an ordered pair of coordinates over a $\langle \mathbb{D}, <\rangle \times \langle \mathbb{D}, <\rangle$ plane, where $\langle \mathbb{D}, <\rangle$ is supposed to be linear. Since the ending point of an interval must be greater than or equal to the starting point, only the north-west half-plane is considered. Clearly, this geometrical interpretation has a good meaning only when HS-formulas are interpreted over linear frames. Here is the standard notation:

- $\diamondsuit\phi \triangleq \langle B\rangle\phi$ (ϕ holds at a point right below the current one).

- $\Diamond_{\uparrow}\phi \triangleq \langle \overline{B} \rangle \phi$ (ϕ holds at a point right above the current one).
- $\Diamond_{\rightarrow}\phi \triangleq \langle E \rangle \phi$ (ϕ holds somewhere to the right of the current point).
- $\Diamond_{\leftarrow}\phi \triangleq \langle \overline{E} \rangle \phi$ (ϕ holds somewhere to the left of the current point).
- $\Diamond_{\updownarrow}\phi \triangleq \Diamond_{\downarrow}\phi \vee \Diamond_{\uparrow}\phi$ (ϕ holds at a point with the same latitude and a different longitude).
- $\Diamond_{\leftrightarrow}\phi \triangleq \Diamond_{\downarrow}\phi \vee \Diamond_{\uparrow}\phi$ (ϕ holds at a point with the same longitude and a different latitude).

Notice that, in order to obtain the mirror image (inverse) of a formula written in the geometrical notation, one should simultaneously replace all $\Diamond_{\downarrow}$ by $\Diamond_{\leftarrow}$ and all $\Diamond_{\uparrow}$ by $\Diamond_{\rightarrow}$, and vice versa. Using this geometrical interpretation, Venema has axiomatized HS over the class of all structures, the class of all linear structures, the class of all discrete structures, and $\mathbb{Q}$. The basic axiomatic system for HS includes the following axioms and their mirror-images:

(A-HS1) enough propositional tautologies;
(A-HS2a) $\Box_{\downarrow}(p \rightarrow q) \rightarrow (\Box_{\downarrow}p \rightarrow \Box_{\downarrow}q)$;
(A-HS2b) $\Box_{\uparrow}(p \rightarrow q) \rightarrow (\Box_{\uparrow}p \rightarrow \Box_{\uparrow}q)$;
(A-HS3a) $\Diamond_{\downarrow}\Diamond_{\downarrow}p \rightarrow \Diamond_{\downarrow}p$;
(A-HS3b) $\Diamond_{\uparrow}\Diamond_{\uparrow}p \rightarrow \Diamond_{\uparrow}p$;
(A-HS4a) $\Diamond_{\downarrow}\Box_{\uparrow}p \rightarrow p$;
(A-HS4b) $\Diamond_{\uparrow}\Box_{\downarrow}p \rightarrow p$;
(A-HS5) $\Diamond_{\downarrow}\top \rightarrow \Diamond_{\downarrow}\Box_{\downarrow}\bot$;
(A-HS6) $\Box_{\downarrow}\bot \rightarrow \Box_{\rightarrow}\bot$;
(A-HS7a) $\Diamond_{\uparrow}\Diamond_{\leftarrow}p \rightarrow \Diamond_{\leftarrow}\Diamond_{\uparrow}p$;
(A-HS7b) $\Diamond_{\downarrow}\Diamond_{\rightarrow}p \leftrightarrow \Diamond_{\rightarrow}\Diamond_{\downarrow}p$;
(A-HS7c) $\Diamond_{\rightarrow}\Diamond_{\uparrow}p \rightarrow \Diamond_{\uparrow}\Diamond_{\rightarrow}p$;
(A-HS8) $(\Diamond_{\downarrow}p \wedge \Diamond_{\downarrow}q) \rightarrow [\Diamond_{\downarrow}(p \wedge \Diamond_{\downarrow}q) \vee \Diamond_{\downarrow}(p \wedge q) \vee \Diamond_{\downarrow}(\Diamond_{\downarrow}p \wedge q)]$,

and the following inference rules: Modus Ponens, Generalization for $\Box_{\downarrow}, \Box_{\uparrow}, \Box_{\rightarrow}$, and $\Box_{\leftarrow}$, and a pair of additional, un-orthodox rules which guarantee that all vertical and horizontal lines in the model are 'syntactically represented':

$$\frac{hor(p) \rightarrow \phi}{\phi} \quad \frac{ver(q) \rightarrow \psi}{\psi},$$

where p, q do not occur in ϕ, ψ respectively, and

- $hor(\phi) \triangleq \phi \wedge \Box_{\leftarrow}\phi \wedge \Box_{\rightarrow}\phi \wedge \Box_{\uparrow}(\neg\phi \wedge \Box_{\leftarrow}\neg\phi \wedge \Box_{\rightarrow}\neg\phi) \wedge \Box_{\downarrow}(\neg\phi \wedge \Box_{\leftarrow}\neg\phi \wedge \Box_{\rightarrow}\neg\phi)$;
- $ver(\phi) \triangleq \phi \wedge \Box_{\uparrow}\phi \wedge \Box_{\downarrow}\phi \wedge \Box_{\leftarrow}(\neg\phi \wedge \Box_{\uparrow}\neg\phi \wedge \Box_{\downarrow}\neg\phi) \wedge \Box_{\rightarrow}(\neg\phi \wedge \Box_{\uparrow}\neg\phi \wedge \Box_{\downarrow}\neg\phi)$.

The formula $hor(\phi)$ holds at an interval $[d_0, d_1]$ if and only if ϕ holds at any $[d_2, d_1]$ where $d_2 \leq d_1$ and nowhere else. Geometrically, it represents a horizontal line on which ϕ is true, and only there. Likewise $ver(\phi)$ says that ϕ is true exactly at the points of some vertical line.

Theorem 10. *The above axiomatic system is sound and complete for the class of all non-strict structures.*

Theorem 11. *A sound and complete axiomatic system for the class of discrete structures can be obtained from the system for the class of all non-strict structures by adding the following axiom:*

(A-HS^{z}) ***discrete****.*

A sound and complete axiomatic system for the class of linear structures can be obtained from the system for the class of all non-strict structures by replacing axiom **(A-HS8)** *by the following axiom:*

(A-HS^{lin}) $(\Diamond_{\downarrow}\Diamond_{\uparrow}p) \rightarrow (\Diamond_{\downarrow}p \vee p \vee \Diamond_{\uparrow}p)$, $(\Diamond_{\uparrow}\Diamond_{\downarrow}p) \rightarrow (\Diamond_{\downarrow}p \vee p \vee \Diamond_{\uparrow}p)$.

A sound and complete axiomatic system for $\mathbb{Q}$ can be obtained from the system for the class of linear structures by adding the following axiom:

(A-HS$^{\mathbb{Q}}$) $\Diamond_{\leftarrow}\top \wedge \Diamond_{\uparrow}\top \wedge$ ***dense***.

In conclusion, we note that, besides D, BE, and A$\overline{\text{A}}$, there exist other interesting fragments of HS, such as B, E, and D$\overline{\text{D}}$, where $\overline{\text{D}}$ is the transpose of D (D$\overline{\text{D}}$ was already mentioned in [59]), which have not been investigated till now. Moreover, at the best of our knowledge, the strict logic HS$^-$ has not been studied yet and thus no complete axiomatic systems and decidability/undecidablity results have been explicitly established for it.

3.2 Interval logics with binary operators

3.2.1 The chop operator and (Local) Propositional Interval Logics. Arguably, the most natural binary interval modality is the *chop* operator C. As proved in [77], such an operator is not definable in HS. The logic that features the operator C and the modal constant π, interpreted according to the non-strict semantics, is the propositional fragment of first-order Interval Temporal Logic (ITL) introduced by Moszkowski in [84] (see Section 4.1). Such a fragment is usually denoted by PITL. PITL-formulas are defined as follows:

$$\phi ::= p \mid \pi \mid \neg\phi \mid \phi \wedge \psi \mid \phi C \psi.$$

The modalities $\langle B \rangle$ and $\langle E \rangle$ are definable in PITL as follows:

- $\langle B \rangle\phi \triangleq \phi C \neg\pi$, and
- $\langle E \rangle\phi \triangleq \neg\pi C \phi$.

As a matter of fact, the study of PITL was originally confined to the class of discrete linear orderings with finite time, with the *chop* operator paired with a **next** operator, denoted by $\bigcirc$, instead of π. For any φ, $\bigcirc\varphi$ holds at a given (discrete) interval $\sigma = s_1 s_2 \ldots s_n$, with $n > 1$, if φ holds at the interval $\sigma' = s_2 \ldots s_n$. It is immediate to see that, over discrete linear orderings, the modal constant π and the *next* operator are inter-changeable. On the one hand, $\pi \triangleq \bigcirc\bot$; on the other hand, for any φ, $\bigcirc\varphi \triangleq l1C\phi$.
The logic PITL is quite expressive, as the following result from [84] testifies (the proof is given in the Appendix).

Theorem 12. *The satisfiability problem for* PITL *interpreted over the class of non-strict discrete structures is undecidable.*

Since PITL is strictly more expressive than both BE and D over the class of discrete linear structures, the above result does not transfer to any of them. On the contrary, the undecidability of the satisfiability problem for PITL over dense structures as well as over all linear structures immediately follows from the undecidability of BE over such structures.

Corollary 2. *The satisfiability problem for* PITL*-formulas interpreted over the class of (non-strict) dense linear structures is undecidable.*

Corollary 3. *The satisfiability problem for* PITL *interpreted over the class of (non-strict) linear structures is undecidable.*

It is worth remarking that the propositional counterpart of the fragment of ITL that only includes the *chop* operator, as far as we know, has not been investigated.

Decidable variants of PITL, interpreted over finite or infinite discrete structures, have been obtained by imposing the so-called *locality projection principle* [84]. Such a locality constraint states that each propositional variable is true over an interval if and only if it is true at its first state. This allows one to collapse all the intervals starting at the same state into the single interval consisting of the first state only.

Let Local PITL (LPITL for short) be logic obtained by imposing the locality projection principle to PITL. The syntax of LPITL coincides with that of PITL, while its semantic clauses are obtained from PITL ones by modifying the truth definition of propositional variables as follows:

(loc-PS1) $\mathbf{M}^+, [d_0, d_1] \Vdash p$ iff $p \in V(d_0)$.

where the valuation function V has been adapted to evaluate propositional variables over points instead of intervals.

Various extensions of LPITL have been proposed in the literature. In [84], Moszkowski focused his attention on the extension of LPITL (over finite time) with quantification over propositional variables, and he proved the decidability of the resulting logic, denoted by QLPITL, by reducing its satisfiability problem to that of QPTL, namely, the point-based Quantified Propositional Temporal Logic, interpreted over discrete linear structures with an initial point (as a matter of fact, QLPITL is translated into QPTL over finite time whose decidability can be proved by a simple adaptation of the standard proof for QPTL over infinite time).

Theorem 13. QPTL *is at least as expressive as* QLPITL *interpreted over the class of (non-strict) discrete linear structures.*

As a consequence, since QPTL is (non-elementarily) decidable, we have the following result.

Corollary 4. *The satisfiability problem for the logic* QLPITL*, interpreted over the class of (non-strict) discrete linear structures is (non-elementarily) decidable.*

From Corollary 4, it immediately follows the (non-elementary) decidability of LPITL. A lower bound for the satisfiability problem for LPITL, and thus for any extension of it, has been given by Kozen (the proof of such a result can be found in [84]).

Theorem 14. *Satisfiability for LPITL is non-elementary.*

In a number of papers [84,86,87,88,89], Moszkowski explored the extension of LPITL with the so-called *chop-star* modality, denoted by *. For any φ, φ^* holds over a given (discrete) interval if and only if the interval can be chopped into zero or more parts such that φ holds over each of them. The resulting logic, that we denote by LPITL*, is interpreted over either finite or infinite discrete linear structures. A sound and complete axiomatic system for LPITL* with finite time is given in [89].

Theorem 15. *The following axiomatic system is sound and complete for the class of (non-strict) discrete linear structures:*

(A-CLPITL1) *enough propositional tautologies;*
(A-CLPITL2) $(\phi C \psi) C \xi \leftrightarrow \phi C (\psi C \xi)$;
(A-CLPITL3) $(\phi \vee \psi) C \xi \rightarrow (\phi C \xi) \vee (\psi C \xi)$;
(A-CLPITL4) $\xi C (\phi \vee \psi) \rightarrow (\xi C \phi) \vee (\xi C \psi)$;
(A-CLPITL5) $\pi C \phi \leftrightarrow \phi$;
(A-CLPITL6) $\phi C \pi \leftrightarrow \phi$;
(A-CLPITL7) $p \rightarrow \neg(\neg p C \top)$, *with* $p \in \mathcal{AP}$;
(A-CLPITL8) $\neg(\neg(\phi \rightarrow \psi) C \top) \wedge \neg(\top C \neg(\xi \rightarrow \chi)) \rightarrow (\phi C \xi) \rightarrow (\psi C \chi)$;
(A-CLPITL9) $\bigcirc \phi \rightarrow \neg \bigcirc \neg \phi$;
(A-CLPITL10) $\phi \wedge \neg(\top C \neg(\phi \rightarrow \neg \bigcirc \neg \phi)) \rightarrow \neg(\top C \neg \phi)$;
(A-CLPITL11) $\phi^* \leftrightarrow \pi \vee (\phi \wedge \bigcirc \top) C \phi^*$,

together with Modus Ponens and the following inference rules:

$$\frac{\phi}{\neg(\top C \neg \phi)}, \quad \frac{\phi}{\neg(\neg \phi C \top)}.$$

All axioms have a fairly natural interpretation. In particular, locality is basically dealt with by Axiom **A-CLPITL7**.

As a matter of fact, the chop-star operator is a special case of a more general operator, called the *projection* operator. Such a binary operator, denoted by `proj`, yields general repetitive behaviour: for any given pair of formulas ϕ, ψ, ϕ `proj` ψ holds over an interval if such an interval can be partitioned into a series of sub-intervals each of which satisfies ϕ and ψ (called the *projected formula*) holds over the new interval formed from the end points of these sub-intervals. Let us denote by LPITL$_{proj}$ the extension of LPITL with the projection operator `proj`. By taking advantage from such an operator, LPITL$_{proj}$ can express meaningful iteration constructs, such as *for* and *while* loops. Furthermore, the chop-star operator can be easily defined in terms of projection operator as follows: $\phi^* \triangleq \phi$ `proj` $\top$. LPITL$_{proj}$ was originally proposed by Moszkowski in [84] and later systematically investigated by Bowman and Thompson [10,11]. In particular, a tableau-based decision procedure and a sound and complete axiomatic system for LPITL$_{proj}$, interpreted over finite discrete structures, is given in [11].

The core of the tableau method is the definition of suitable normal forms for all operators of the logic. These normal forms provide inductive definitions of the operators. Then, in the style of [112], a tableau decision procedure to check satisfiability of LPITL$_{proj}$ formulas is established. (Although the method has been developed at the propositional level, the authors advocate its validity also for first-order LPITL$_{proj}$.)
The normal form for LPITL$_{proj}$ formulas has the following general format:

$$(\pi \wedge \phi_e) \vee \bigvee_i (\phi_i \wedge \bigcirc \phi'_i)$$

where ϕ_e and ϕ_i are point formulas, that is, formulas that are evaluated at single points, and ϕ'_i is an arbitrary LPITL$_{proj}$ formula. The first disjunct states when a formula is satisfied over a point interval, while the second one states the possible ways in which a formula can be satisfied over a strict interval, namely, a point formula must hold at the initial point and then an arbitrary formula must hold over the remainder of the interval. It is worth noting that this normal form embodies a recipe for evaluating LPITL_{proj} formulas: the first disjunct is the base case, while the second disjunct is the inductive step. Bowman and Thomson showed that any LPITL_{proj} formula can be equivalently transformed into this normal form.

In [11], Bowman and Thomson also provided a sound and complete axiomatic system for LPITL_{proj}, interpreted over discrete linear structures. Let ϕ, ψ, ξ be arbitrary formulas and $p \in \mathcal{AP}$. The proposed system includes the following axioms:

(A-LPITL1) enough propositional tautologies;
(A-LPITL2) $\neg\pi \leftrightarrow \bigcirc\top$;
(A-LPITL3) $\bigcirc\phi \rightarrow \neg \bigcirc \neg\phi$;
(A-LPITL4) $\bigcirc(\phi \rightarrow \psi) \rightarrow \bigcirc\phi \rightarrow \bigcirc\psi$;
(A-LPITL5) $(\bigcirc\phi)C\psi \leftrightarrow \bigcirc(\phi C\psi)$;
(A-LPITL6) $(\phi \vee \psi)C\xi \leftrightarrow \phi C\xi \vee \phi C\xi$;
(A-LPITL7) $\phi C(\psi \vee \xi) \leftrightarrow \phi C\psi \vee \phi C\xi$;
(A-LPITL8) $\phi C(\psi C\xi) \leftrightarrow (\phi C\psi)C\xi$;
(A-LPITL9) $(p \wedge \phi)C\phi \leftrightarrow p \wedge (\phi C\psi)$, with $p \in \mathcal{AP}$;
(A-LPITL10) $\pi C\phi \leftrightarrow \phi C\pi \leftrightarrow \phi$;
(A-LPITL11) $\phi \; proj \; \pi \leftrightarrow \pi$;
(A-LPITL12) $\phi \; proj \; (\psi \vee \xi) \leftrightarrow (\phi \; proj \; \psi) \vee (\phi \; proj \; \xi)$;
(A-LPITL13) $\phi \; proj \; (p \wedge \psi) \leftrightarrow p \wedge (\phi \; proj \; \psi)$;
(A-LPITL14) $\phi \; proj \; \bigcirc\psi \leftrightarrow (\phi \wedge \neg\pi)C(\phi \; proj \; \psi)$.

The inference rules, besides Modus Ponens and $\bigcirc$-generalization, include the following rule:

$$\frac{\phi \rightarrow \bigcirc^k\phi}{\neg\phi}.$$

Theorem 16. *The above axiomatic system is sound and complete for the class of (non-strict) discrete structures.*

Finally, Kono [72] presents a tableau-based decision procedure for QLPITL with *projection*, which has been successfully implemented. The method generates a deterministic state diagram as a verification result. Although the associated axiomatic system is probably unsound (see [89]), Kono's work actually inspired Bowman and Thompson's one.

3.2.2. The logics CDT and BCDT$^+$. The most expressive propositional interval logic over (non-strict) linear orderings proposed in the literature is Venema's CDT [110]. A generalization of CDT to (non-strict) partial orderings with the linear intervals property, called BCDT$^+$ has been recently investigated by Goranko, Montanari, and Sciavicco [48]. The language of CDT and BCDT$^+$ contains the three binary operators C, D, and T, together with the modal constant π. Formulas of CDT are generated by the following abstract grammar:

$$\phi ::= \pi \mid p \mid \neg\phi \mid \phi \wedge \psi \mid \phi C\psi \mid \phi D\psi \mid \phi T\psi.$$

The semantics of both CDT and BCDT$^+$ are non-strict.

As for the expressive power, Venema compared CDT ability of defining binary operators with that of the fragment $\text{FO}_3[<](x_i, x_j)$ of first-order logic over linear orderings with at most three variables, say x_1, x_2, and x_3, among which at most x_i, x_j, with $i, j \in \{1, 2, 3\}$, are free [110]. He proves the following result (a sketch of the proof is given in the Appendix).

Theorem 17. *Every binary modal operator definable in* $FO_3[<](x_i, x_j)$ *has an equivalent in* CDT*, and vice versa.*

As for the relationships with the other propositional interval logics, interpreted over linear orderings, CDT is strictly more expressive than PITL, since the latter is not able to access any interval which is not a sub-interval of the current interval. Moreover, it is immediate to show that CDT subsumes HS:

- $\Diamond_{r}\phi = (\neg\pi)C\phi$;
- $\Diamond_{l}\phi = (\neg\pi)D\phi$;
- $\Diamond_{u}\phi = (\neg\pi)T\phi$;
- $\Diamond_{d}\phi = \phi C(\neg\pi)$.

A sound and complete axiomatic system for CDT over (non-strict) linear structures has been defined by Venema in [110]. Let us define $hor(\phi)$ as in the case of HS. The axiomatic system for CDT includes the following axioms, and their inverses (obtained by exchanging the arguments of all C occurrences, and replacing each occurrence of T by D and vice versa):

(A-CDT1) enough propositional tautologies;
(A-CDT2a) $(\phi \vee \psi)C\xi \leftrightarrow \phi C\xi \vee \psi C\xi$;
(A-CDT2b) $(\phi \vee \psi)T\xi \leftrightarrow \phi T\xi \vee \psi T\xi$;
(A-CDT2c) $\phi T(\psi \vee \xi) \leftrightarrow \phi T\psi \vee \phi T\xi$;
(A-CDT3a) $\neg(\phi T\psi)C\phi \rightarrow \neg\psi$;
(A-CDT3b) $\neg(\phi T\psi)D\psi \rightarrow \neg\phi$;
(A-CDT3c) $\phi T\neg(\psi C\phi) \rightarrow \neg\psi$;
(A-CDT4) $\neg\pi C\top \leftrightarrow \neg\pi$;
(A-CDT5a) $\pi C\phi \leftrightarrow \phi$;
(A-CDT5b) $\pi T\phi \leftrightarrow \phi$;
(A-CDT5c) $\phi T\pi \leftrightarrow \phi$;
(A-CDT6) $[(\pi \wedge \phi)C\top \wedge ((\pi \wedge \psi)C\top)C\top] \rightarrow (\pi \wedge \psi)C\top$;
(A-CDT6a) $(\phi C\psi)C\xi \leftrightarrow \phi C(\psi C\xi)$;
(A-CDT6b) $\phi T(\psi C\xi) \leftrightarrow (\psi C(\phi T\xi) \vee (\xi T\phi)T\psi)$;
(A-CDT6c) $\psi C(\phi T\xi) \rightarrow \phi T(\psi C\xi)$;
(A-CDT7d) $(\phi T\psi)C\xi \rightarrow ((\xi D\phi)T\psi \vee \psi C(\phi D\xi))$;

and the following derivation rules: Modus Ponens, Generalization:

$$\frac{\phi}{\neg(\neg\phi C\psi)}, \quad \frac{\phi}{\neg(\neg\varphi T\psi)}, \quad \frac{\phi}{\neg(\psi T\neg\phi)}, \text{ and their inverses,}$$

and the Consistency rule: if $p \in \mathcal{AP}$ and p does not occur in ϕ, then

$$\frac{\text{hor}(p) \rightarrow \phi}{\phi}.$$

Theorem 18. *The above axiomatic system is sound and complete for the class of (non-strict) linear orderings.*

Theorem 19. *A sound and complete axiomatic system for the class of (non-strict) dense linear orderings can be obtained from the system for the class of (non-strict) linear orderings by adding the following axiom:*

(A-CDT^d) $\neg\pi \rightarrow (\neg\pi C \neg\pi)$.

A sound and complete axiomatic system for the class of (non-strict) discrete linear orderings can be obtained from the system for the class of (non-strict) linear orderings by adding the following axiom:

(A-CDT^z) $\pi \vee (l1C\top) \wedge (\top Cl1)$*;*

A sound and complete axiomatic system for $\mathbb{Q}$ *can be obtained from the system for the class of (non-strict) linear orderings by adding the following axiom:*

(A-CDT$^{\mathbb{Q}}$) $(\neg\pi \rightarrow (\neg\pi C \neg\pi)) \wedge (\neg\pi T\top) \wedge (\neg\pi D\top)$.

In [110], Venema also developed a sound and complete natural deduction system for CDT, similar to the natural deduction system for relation algebras earlier developed by Maddux [75].

Finally, as a consequence from previous results for HS and PITL, the satisfiability (resp. validity) for CDT is not decidable over almost all interesting classes of linear orderings, including all, dense, discrete, etc. Again, the strict versions of CDT and BCDT$^+$ have not been explicitly studied yet, but it is natural to expect that similar results apply there, too.

3.3 A general tableau method for propositional interval logics

In this section we describe a sound and complete tableau method for BCDT$^+$, developed by Goranko, Montanari and Sciavicco in [48], which combines features of tableau methods for modal logics with constraint label management and the classical tableau method for first-order logic. The proposed method can be adapted to variations and subsystems of BCDT$^+$, thus providing a general tableau method for propositional interval logics.

First, some basic terminology. A **finite tree** is a finite directed connected graph in which every node, apart from one (the **root**), has exactly one incoming arc. A **successor** of a node $\mathbf{n}$ is a node $\mathbf{n'}$ such that there is an edge from $\mathbf{n}$ to $\mathbf{n'}$. A **leaf** is a node with no successors; a **path** is a sequence of nodes $\mathbf{n_0}, \ldots, \mathbf{n_k}$ such that, for all $i = 0, \ldots, k-1$, $\mathbf{n_{i+1}}$ is a successor of $\mathbf{n_i}$; a **branch** is a path from the root to a leaf. The **height** of a node $\mathbf{n}$ is the maximum length (number of edge) of a path from $\mathbf{n}$ to a leaf. If $\mathbf{n}, \mathbf{n'}$ belong to the same branch and the height of $\mathbf{n}$ is less than or equal to the height of $\mathbf{n'}$, we write $\mathbf{n} \prec \mathbf{n'}$. Let $\langle \mathbb{C}, < \rangle$ be a finite partial order. A **labeled formula**, with label in $\mathbb{C}$, is a pair $(\phi, [c_i, c_j])$, where $\phi \in \mathrm{BCDT}^+$ and $[c_i, c_j] \in \mathbb{I}(\mathbb{C})^+$.
For a node $\mathbf{n}$ in a tree, the **decoration** $\nu(\mathbf{n})$ is a triple $((\phi, [c_i, c_j]), \mathbb{C}, u_{\mathbf{n}})$, where $\langle \mathbb{C}, < \rangle$ is a finite partial order, $(\phi, [c_i, c_j])$ is a labeled formula, with label in $\mathbb{C}$, and $u_{\mathbf{n}}$ is a **local flag function** which associates the values 0 or 1 with every branch B containing $\mathbf{n}$. Intuitively, the value 0 for a node n with respect to a branch B means that n can be expanded on B. For the sake of simplicity, we will often assume the interval $[c_i, c_j]$ to consist of the elements $c_i < c_{i+1} < \cdots < c_j$, and sometimes, with a little abuse of notation, we will write $\mathbb{C} = \{c_i < c_k, c_m < c_j, \ldots\}$. A **decorated tree** is a tree in which every node has a decoration $\nu(\mathbf{n})$. For every decorated tree, we define a **global flag function** u acting on pairs *(node, branch through that node)* as $u(\mathbf{n}, B) = u_{\mathbf{n}}(B)$. Sometimes, for convenience, we will include in the decoration of the nodes the global flag function instead of the local ones. For any branch B in a decorated tree, we denote by $\mathbb{C}_B$ the ordered set in the decoration of the leaf B, and for any node $\mathbf{n}$ in a decorated tree, we denote by $\Phi(\mathbf{n})$ the formula in its

decoration. If B is a branch, then $B \cdot \mathbf{n}$ denotes the result of the expansion of B with the node $\mathbf{n}$ (addition of an edge connecting the leaf of B to $\mathbf{n}$). Similarly, $B \cdot \mathbf{n_1} \mid \ldots \mid \mathbf{n_k}$ denotes the result of the expansion of B with k immediate successor nodes $\mathbf{n_1}, \ldots, \mathbf{n_k}$ (which produces k branches extending B). A tableau for BCDT$^+$ will be defined as a special decorated tree. We note again that $\mathbb{C}$ remains finite throughout the construction of the tableau.

Definition 1. *Given a decorated tree $\mathcal{T}$, a branch B in $\mathcal{T}$, and a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\phi, [c_i, c_j]), \mathbb{C}, u)$, with $u(\mathbf{n}\ , B) = 0$, the* ***branch-expansion rule*** *for B and $\mathbf{n}$ is defined as follows (in all the considered cases, $u(\mathbf{n'}\ , B') = 0$ for all new pairs $(\mathbf{n'}\ , B')$ of nodes and branches).*

- *If $\phi = \neg\neg\psi$, then expand the branch to $B \cdot \mathbf{n_0}$, with $\nu(\mathbf{n_0}) = ((\psi,\ [c_i, c_j]),\ \mathbb{C}_B, u)$.*
- *If $\phi = \psi_0 \wedge \psi_1$, then expand the branch to $B \cdot \mathbf{n_0}\ \cdot \mathbf{n_1}$, with $\nu(\mathbf{n_0}) = ((\psi_0, [c_i, c_j]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_1}) = ((\psi_1, [c_i, c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg(\psi_0 \wedge \psi_1)$, then expand the branch to $B \cdot \mathbf{n_0} | \mathbf{n_1}$, with $\nu(\mathbf{n_0}) = ((\neg\psi_0, [c_i, c_j]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_1}) = ((\neg\psi_1, [c_i, c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg(\psi_0 C \psi_1)$ and c is the least element of $\mathbb{C}_B$, with $c_i \leq c \leq c_j$, which has not been used yet to expand the node $\mathbf{n}$ on B, then expand the branch to $B \cdot \mathbf{n_0} | \mathbf{n_1}$, with $\nu(\mathbf{n_0})$ $= ((\neg\psi_0, [c_i, c]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_1}) = ((\neg\psi_1, [c, c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg(\psi_0 D \psi_1)$, c is a minimal element of $\mathbb{C}_B$ such that $c \leq c_i$, and there exists $c' \in [c, c_i]$ which has not been used yet to expand the node $\mathbf{n}$ on B, then take the least such $c' \in [c, c_i]$ and expand the branch to $B \cdot \mathbf{n_0} | \mathbf{n_1}$, with $\nu(\mathbf{n_0}) = ((\neg\psi_0, [c', c_i]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_1}) = ((\neg\psi_1, [c', c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg(\psi_0 T \psi_1)$, c is a maximal element of $\mathbb{C}_B$ such that $c_j \leq c$, and there exists $c' \in [c_j, c]$ which has not been used yet to expand the node $\mathbf{n}$ on B, then take the greatest such $c' \in [c_j, c]$ and expand the branch to $B \cdot \mathbf{n_0} | \mathbf{n_1}$, so that $\nu(\mathbf{n_0}) = ((\neg\psi_0, [c_j, c']), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_1}) = ((\neg\psi_1, [c_i, c']), \mathbb{C}_B, u)$.*
- *If $\phi = (\psi_0 C \psi_1)$, then expand the branch to $B \cdot (\mathbf{n_i} \cdot \mathbf{m_i}) | \ldots | (\mathbf{n_j} \cdot \mathbf{m_j}) | (\mathbf{n'_i} \cdot \mathbf{m'_i}) | \ldots | (\mathbf{n'_{j-1}} \cdot \mathbf{m'_{j-1}})$, where:*
  1. *for all $c_k \in [c_i, c_j]$, $\nu(\mathbf{n_k}) = ((\psi_0, [c_i, c_k]), \mathbb{C}_B, u)$ and $\nu(\mathbf{m_k}) = ((\psi_1,\ [c_k, c_j]),\ \mathbb{C}_B, u)$;*
  2. *for all $i \leq k \leq j-1$, let $\mathbb{C}_k$ be the interval structure obtained by inserting a new element c between c_k and c_{k+1} in $[c_i, c_j]$, $\nu(\mathbf{n'_k}) = ((\psi_0, [c_i, c]), \mathbb{C}_k, u)$, and $\nu(\mathbf{m'_k})$ $= ((\psi_1, [c, c_j]), \mathbb{C}_k, u)$.*
- *If $\phi = (\psi_0 D \psi_1)$, then repeatedly expand the current branch, once for each minimal element c (where $[c, c_i] = \{c = c_0 < c_1 < \cdots c_i\}$), by adding the decorated sub-tree $(\mathbf{n_0} \cdot \mathbf{m_0}) | \ldots | (\mathbf{n_i} \cdot \mathbf{m_i}) | (\mathbf{n'_1} \cdot \mathbf{m'_1}) | \ldots | (\mathbf{n'_i} \cdot \mathbf{m'_i}) | (\mathbf{n''_0} \cdot \mathbf{m''_0}) | \ldots | (\mathbf{n''_i} \cdot \mathbf{m''_i})$ to its leaf, where:*
  1. *for all c_k such that $c_k \in [c, c_i]$, $\nu(\mathbf{n_k}) = ((\psi_0, [c_k, c_j]), \mathbb{C}_B, u)$ and $\nu(\mathbf{m_k}) = ((\psi_1, [c_k, c_i]), \mathbb{C}_B, u)$;*
  2. *for all $0 < k \leq i$, let $\mathbb{C}_k$ be the interval structure obtained by inserting a new element c' immediately before c_k in $[c, c_i]$, and $\nu(\mathbf{n'_k}) = ((\psi_0, [c', c_i]),\ \mathbb{C}_k, u)$ and $\nu(\mathbf{m'_k}) = ((\psi_1,\ [c', c_j]), \mathbb{C}_k, u)$;*
  3. *for all $0 \leq k \leq i$, let $\mathbb{C}_k$ be the interval structure obtained by inserting a new element c' in $\mathbb{C}_B$, with $c' < c_k$, which is incomparable with all existing predecessors of c_k, $\nu(\mathbf{n''_k}) = ((\psi_0, [c', c_i]), \mathbb{C}_k, u)$, and $\nu(\mathbf{m''_k}) = ((\psi_1, [c', c_j]), \mathbb{C}_k, u)$.*
- *If $\phi = (\psi_0 T \psi_1)$, then repeatedly expand the current branch, once for each maximal element c (where $[c_j, c] = \{c_j < c_{j+1} < \cdots c_n = c\}$), by adding the decorated sub-tree $(\mathbf{n_j} \cdot \mathbf{m_j}) | \ldots | (\mathbf{n_n} \cdot \mathbf{m_n}) | (\mathbf{n'_j} \cdot \mathbf{m'_j}) | \ldots | (\mathbf{n'_{n-1}} \cdot \mathbf{m'_{n-1}}) | (\mathbf{n''_j} \cdot \mathbf{m''_j}) | \ldots | (\mathbf{n''_n} \cdot \mathbf{m''_n})$ to its leaf, where:*

1. *for all c_k such that $c_k \in [c_j, c]$, $\nu(\mathbf{n_k}) = ((\psi_0, [c_j, c_k]), \mathbb{C}_B, u)$ and $\nu(\mathbf{m_k}) = ((\psi_1, [c_i, c_k]), \mathbb{C}_B, u)$;*
2. *for all $j \leq k < n$, let $\mathbb{C}_k$ be the interval structure obtained by inserting a new element c' immediately after c_k in $[c_j, c]$, and $\nu(\mathbf{n'_k}) = ((\psi_0, [c_j, c']), \mathbb{C}_k, u)$ and $\nu(\mathbf{m'_k}) = ((\psi_1, [c_i, c']), \mathbb{C}_k, u)$;*
3. *for all $j \leq k \leq n$, let $\mathbb{C}_k$ be the interval structure obtained by inserting a new element c' in $\mathbb{C}_B$, with $c_k < c'$, which is incomparable with all existing successors of c_k, $\nu(\mathbf{n''_k}) = ((\psi_0, [c_j, c']), \mathbb{C}_k, u)$, and $\nu(\mathbf{m''_k}) = ((\psi_1, [c_i, c']), \mathbb{C}_k, u)$.*

Finally, for any node $\mathbf{m}$ ($\neq \mathbf{n}$) in B and any branch B' extending B, let $u(\mathbf{m}, B')$ be equal to $u(\mathbf{m}, B)$, and for any branch B' extending B, $u(\mathbf{n}, B') = 1$, unless $\phi = \neg(\psi_0 C \psi_1)$, $\phi = \neg(\psi_0 D \psi_1)$, or $\phi = \neg(\psi_0 T \psi_1)$ (in such cases $u(\mathbf{n}, B') = 0$).

Let us briefly explain the expansion rules for $\psi_0 C \psi_1$ and $\neg(\psi_0 C \psi_1)$ (similar considerations hold for the other temporal operators). The rule for the (existential) formula $\psi_0 C \psi_1$ deals with the two possible cases: either there exists c_k in $\mathbb{C}_B$ such that $c_i \leq c_k \leq c_j$ and ψ_0 holds over $[c_i, c_k]$ and ψ_1 holds over $[c_k, c_j]$ or such an element c_k must be added. The (universal) formula $\neg(\psi_0 C \psi_1)$ states that, for all $c_i \leq c \leq c_j$, ψ_0 does not hold over $[c_j, c]$ or ψ_1 does not hold over $[c, c_j]$. As a matter of fact, the expansion rule imposes such a condition for a single element c in $\mathbb{C}_B$ (the least element which has not been used yet), and it does not change the flag (which remains equal to 0). In this way, all elements will be eventually taken into consideration, including those elements in between c_i and c_j that will be added to $\mathbb{C}_B$ in some subsequent steps of the tableau construction.
Let us define now the notions of open and closed branch. We say that a node $\mathbf{n}$ in a decorated tree $\mathcal{T}$ is **available on a branch** B to which it belongs if and only if $u(\mathbf{n}, B) = 0$. The branch-expansion rule is **applicable** to a node $\mathbf{n}$ on a branch B if the node is available on B and the application of the rule generates at least one successor node with a new labeled formula. This second condition is needed to avoid looping of the application of the rule on formulas $\neg(\psi_0 C \psi_1), \neg(\psi_0 D \psi_1)$, and $\neg(\psi_0 T \psi_1)$.

Definition 2. *A branch B is **closed** if some of the following conditions holds:*
(i) there are two nodes $\mathbf{n}, \mathbf{n'} \in B$ such that $\nu(\mathbf{n}) = ((\psi, [c_i, c_j]), \mathbb{C}, u)$ and $\nu(\mathbf{n'}) = ((\neg\psi, [c_i, c_j]), \mathbb{C}', u)$ for some formula ψ and $c_i, c_j \in \mathbb{C} \cap \mathbb{C}'$;
(ii) there is a node $\mathbf{n}$ such that $\nu(\mathbf{n}) = ((\pi, [c_i, c_j]), \mathbb{C}, u)$ and $c_i \neq c_j$; or
(iii) there is a node $\mathbf{n}$ such that $\nu(\mathbf{n}) = ((\neg\pi, [c_i, c_j]), \mathbb{C}, u)$ and $c_i = c_j$.
*If none of the above conditions hold, the branch is **open**.*

Definition 3. *The **branch-expansion strategy** for a branch B in a decorated tree $\mathcal{T}$ is defined as follows:*

1. *Apply the branch-expansion rule to a branch B only if it is open;*
2. *If B is open, apply the branch-expansion rule to the closest to the root available node in B for which the branch-expansion rule is applicable.*

Definition 4. *A **tableau** for a given formula $\phi \in$BCDT$^+$ is any finite decorated tree $\mathcal{T}$ obtained by expanding the three-node decorated tree built up from an empty-decoration root and two leaves with decorations $((\phi, [c_b, c_e]), \{c_b < c_e\}, u)$ and $((\phi, [c_b, c_b]), \{c_b\}, u)$, where the value of u is 0, through successive applications of the branch-expansion strategy to the existing branches.*

It is easy to show that if $\phi \in \mathrm{BCDT}^+$, $\mathcal{T}$ is a tableau for ϕ, $\mathbf{n} \in \mathcal{T}$, and $\mathbb{C}$ is the ordered set in the decoration of $\mathbf{n}$, then $\langle \mathbb{C}, < \rangle$ is an interval structure.

Theorem 20 (soundness and completeness). *If $\phi \in$BCDT$^+$ and a tableau $\mathcal{T}$ for ϕ is closed, then ϕ is not satisfiable. Moreover, if $\phi \in$ BCDT$^+$ is valid formula, then there is a closed tableau for $\neg\phi$.*

3.4 Restricted interval logics: split logics

The aim of restricted interval logics is to find powerful decidable propositional interval logics without resorting to the locality principle. We briefly review the basic features of Split Logic (SL for short) and the achieved results. SLs have been studied by Montanari, Sciavicco, and Vitacolonna in [82], where precise definitions and proofs can be found.
Split logics (SLs for short) are propositional interval logics equipped with operators borrowed from HS and CDT, but interpreted over specific structures, called **split-structures**. Models based on split structures, are called **split models**. A split structure is said to have **maximal intervals** if and only if, for every interval $[d_0, d_1]$ there exists an interval $[d_2, d_3]$ such that $[d_0, d_1]$ is contained in $[d_2, d_3]$ (following the Allen's terminology) and such that there is no interval containing $[d_2, d_3]$.

The abstract syntax defining formulas of split logics is:

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \langle D\rangle\varphi \mid \langle\overline{D}\rangle\varphi \mid \langle F\rangle\varphi \mid \langle\overline{F}\rangle\varphi \mid \varphi C\varphi \mid \varphi D\varphi \mid \varphi T\varphi.$$

The semantic clauses of the modalities are the already given ones for the operators $\langle D\rangle, C, D, T$, plus the following ones for the new operators:

($\langle\overline{D}\rangle$) $\mathbf{M}, [d_0, d_1] \Vdash \langle\overline{D}\rangle\phi$ iff there exist d_2, d_3 such that $d_2 < d_0$, $d_1 < d_3$, and $\mathbf{M}, [d_2, d_3] \Vdash \phi$;
($\langle F\rangle$) $\mathbf{M}, [d_0, d_1] \Vdash \langle F\rangle\phi$ iff there exist d_2, d_3 such that $d_1 < d_2$, $d_2 < d_3$, and $\mathbf{M}, [d_2, d_3] \Vdash \phi$;
($\langle\overline{F}\rangle$) $\mathbf{M}, [d_0, d_1] \Vdash \langle\overline{F}\rangle\phi$ iff there exist d_2, d_3 such that $d_2 < d_0$, $d_3 < d_2$, and $\mathbf{M}, [d_3, d_2] \Vdash \phi$.

The modal constant π can also be introduced.

The interesting feature of split logics is that they can be interpreted over classes of split structures with additional properties that can be translated into granular structures (cfr. [78]). Granular structures (or **layered structures**) are discrete linear point-based structures bounded in the past and infinite in the future, in which a universe of domains replaces the single 'flat' temporal domain. These domains are correlated through **granular primitives** that relate points belonging to the same domain as well as points belonging to different domains. A formal definition of these structures can be found in [78,43]. Intuitively, the picture is as follows: the domain $\bigcup_i T^i$ of layered structures consists of (possibly infinitely) many copies T^i of $\mathbb{N}$, each one being a **layer** of the structure. If there is a finite number n of layers, the structure is called **n-layered** (n-LS), otherwise, the structure is called **ω-layered**. It can be **upward unbounded** (UULS) if there is a finest domain and an infinite sequence of coarser domains, or it can be **downward unbounded** (DULS) if there is a coarsest domain and an infinite sequence of finer ones. In every case, layers are totally ordered according to their degree of 'coarseness' or 'fineness', and each point in a layer is associated with k points in the immediately finer layer (**k-refinability**). This accounts for a view of these structures also as infinite sequences of (possibly infinite) complete k-ary trees.

In the case of UULSs, there is only one infinite tree built up from leaves, which form its first layer. In the case of n-LSs and DULSs, the infinite sequence of respectively finite and infinite trees is ordered according to the ordering of the roots, which form their first layer. In [78,43] monadic second-order theories of granular structures have been studied and their decidability has been proved. Here we are interested in the first-order fragments of those theories, namely, $\mathrm{MFO}[<_1, <_2, \{\downarrow_i\}_{0\leq i<k}]$ interpreted over k-refinable n-LSs and DULSs, and $\mathrm{MFO}[<_2, \{\downarrow_i\}_{0\leq i<k}]$ interpreted over k-refinable UULSs. The symbols in the square brackets are (pre)interpreted as follows: for $0 \leq i < k$, $\downarrow_i (x, y)$ is a binary relation such that y is the i-th point in the refinement of x; $<_1$ is a strict partial order such that $x <_1 y$ when x is in a tree preceding the tree containing y; $x <_2 y$ holds when y is a descendant of x.
Now we consider three possible interpretations for a split logic, namely over (i) the class of bounded below, unbounded above, dense, and with maximal intervals split structures, (ii) the class of bounded below, unbounded above, discrete split structures, and (iii) the class of bounded below, unbounded above, discrete, and with maximal intervals split structures. Any of the above classes of structures corresponds to a particular class of granular 2-refinable structures.

Theorem 21.

1. SL *interpreted over the class of bounded below, unbounded above, dense, and with maximal intervals split structures can be embedded into* $\mathrm{MFO}[<_1, <_2, \{\downarrow_i\}_{0\leq i<k}]$ *interpreted over* DULS*s;*
2. SL *interpreted over the class of bounded below, unbounded above, discrete, and with maximal intervals split structures can be embedded into* $\mathrm{MFO}[<_1, <_2, \{\downarrow_i\}_{0\leq i<k}]$ *interpreted over* n-LS*s;*
3. SL *interpreted over the class of bounded below, unbounded above, discrete split structures can be embedded into the logic* $\mathrm{MFO}[<_2, \{\downarrow_i\}_{0\leq i<k}]$ *interpreted over* UULS*s.*

By exploiting the decidability of such monadic first-order theories over the considered granular structures, the following results can be obtained.

Corollary 5. *The satisfiability problem for* SL*-formulas interpreted over the classes of split structures considered above is decidable.*

4 First-Order Interval Logics and Duration Calculi

Research on interval temporal logics in computer science was originally motivated by problems in the field of specification and verification of hardware protocols, rather than by abstract philosophical or logical issues. Not surprisingly, it focused on first-order, rather than propositional, interval logics. In this section, we summarize some of the most-important developments in first-order interval logics and duration calculi, referring the interested reader to respectively [89] and [61] for more details.

4.1 The logic ITL

First-order ITL, interpreted over discrete linear orderings with finite time intervals, was originally developed by Halpern, Manna, and Moszkowski in [84,58]. The language of ITL includes terms, predicates, Boolean connectives, first-order quantifiers, and the temporal modalities C and $\bigcirc$. Terms are built on variables, constants, and function symbols in the usual way. Constants and function symbols are classified as *global* (or *rigid*), when their interpretation does not vary with time, and *temporal* (or *flexible*), when their interpretation may change over time. Terms are usually denoted by $\theta_1, \ldots, \theta_n$. Predicate symbols are also partitioned into global and temporal ones. They are denoted by $p^i, q^j, \ldots$, where p^i is a predicate of arity i, q^j is a predicate of arity j, and so on. The abstract syntax of ITL formulas is:

$$\phi ::= \theta \mid p^n(\theta_1, \ldots, \theta_n) \mid \exists x\phi | \ \neg\phi \mid \phi \wedge \psi \mid \ \bigcirc \phi \mid \phi C \psi.$$

The semantics of ITL-formulas is a combination of the standard semantics of a first-order temporal logic with the semantics of PITL. An account of possible uses and applications is e.g. [85].

In [37] Dutertre studies the fragment of ITL which we will denote here by ITL_D, involving only the *chop* operator. First, ITL_D is considered over abstract, Kripke-style models $\mathbf{M}^+ = \langle W, R, I\rangle$, where W is a set of worlds (abstract intervals), R is a ternary relation corresponding to Venema's A, and I is a first-order interpretation. Further, Dutertre considers a more concrete semantics, over interval structures with associated 'length' measure represented by a special temporal variable l which takes values in a commutative group $\langle \mathbb{D}, +, -, 0\rangle$. The language is assumed to have the flexible constant l, and the rigid symbols 0 and $+$, respectively interpreted as the neutral element and the addition in $\langle \mathbb{D}, +, 0\rangle$. The semantics of ITL_D-formulas is a combination of the semantics of ITL (without *next*), and the interpretation of l in a model $\mathbf{M}^+$ for an interval $[d_0, d_1]$ is $d_1 - d_0$.

As for the expressive power of ITL_D, note that by means of l one can easily define the modal constant $\pi \triangleq (l = 0)$. So, the HS modalities corresponding to *begin* and *end* are also definable in the language, thus, by the results of the Section 3.1.2, this means that ITL_D is at least as expressive as PITL. The undecidability of this logic is an easy consequence of the above considerations.

Dutertre has provided an axiomatic system for ITL_D, the soundness and completeness proof for which can be found in [37]. In addition to the standard axioms of first-order classical logic, incl. the axioms of identity, and the axioms describing the properties for the temporal domain $\mathbb{D}$, Dutertre's systems involves the following specific axioms for ITL_D:

(A-ITL1) $(\phi C\psi) \wedge \neg(\phi C\xi) \rightarrow (\phi C(\psi \wedge \neg\xi)$;
(A-ITL2) $(\phi C\psi) \wedge \neg(\xi C\psi) \rightarrow (\phi \wedge \neg\xi)C\psi$;
(A-ITL3) $((\phi C\psi); \xi) \leftrightarrow (\phi C(\psi C\xi))$;
(A-ITL4) $(\phi C\psi) \rightarrow \phi$ if ϕ is a rigid formula;
(A-ITL5) $(\phi C\psi) \rightarrow \psi$ if ψ is a rigid formula;
(A-ITL6) $((\exists x)\phi C\psi) \rightarrow (\exists x)(\phi C\psi)$ if x is not free in ψ;
(A-ITL7) $(\phi C(\exists x)\psi) \rightarrow (\exists x)(\phi C\psi)$ if x is not free in ϕ;
(A-ITL8) $((l = x)C\phi) \rightarrow \neg((l = x)C\phi)$;
(A-ITL9) $(\phi C(l = x)) \rightarrow \neg(\neg\phi C(l = x))$;
(A-ITL10) $(l = x + y) \leftrightarrow ((l = x)C(l = y))$;
(A-ITL11) $\phi \rightarrow (\phi C(l = 0))$;
(A-ITL12) $\phi \rightarrow ((l = 0)C\phi)$.

The inference rules are Modus Ponens, Generalization, Necessitation, and the following Monotonicity rule:

$$\frac{\phi \rightarrow \psi}{\phi C\xi \rightarrow \psi C\xi},$$

and the symmetric one. It should be noted that certain restrictions apply to the instantiation with flexible terms in quantified formulas.

As in the propositional case, variants of ITL obtained by imposing the locality constraint have been explored in the literature. In particular, sound and complete axiomatic systems for local variants of ITL (LITL for short) have been developed in [37,38,88].

4.2 The logic NL

The logic ITL has an intrinsic limitation: its modalities do not allow one to 'look' outside the current interval (modalities with this characteristic are called *contracting* modalities). To overcome such a limitation, Zhou and Hansen [116] proposed the first-order logic of *left* and *right* neighbourhood modalities, called *neighbourhood logic* (NL for short), whose propositional fragment has been analyzed in Section 3.1.3.
First-order syntactic features are as in the ITL case. Right and left neighbourhood modalities are denoted by $\Diamond_r$ and $\Diamond_l$, respectively. The abstract syntax of NL formulas is:

$$\phi ::= \theta \mid p^n(\theta_1, \ldots, \theta_n) \mid \exists x\phi \mid \ \neg\phi \mid \phi \wedge \psi \mid \Diamond_l\phi \mid \Diamond_r\phi.$$

where terms $(\theta_1, \ldots, \theta_n)$ are defined as in ITL.
As in the propositional case, the neighbourhood modalities are interpreted in non-strict structures by means of the following clauses:

(LFS1) $\mathbf{M}^+, [d_0, d_1] \Vdash \Diamond_r\phi$ iff there exists d_2 such that $d_1 \leq d_2$ and $\mathbf{M}^+, [d_1, d_2] \Vdash \phi$;
(LFS2) $\mathbf{M}^+, [d_0, d_1] \Vdash \Diamond_r\phi$ iff there exists d_2 such that $d_2 \leq d_0$ and $\mathbf{M}^+, [d_2, d_0] \Vdash \phi$.

The rest of the semantics of NL is defined exactly as in the ITL case. While practically meant to be the ordered additive group of the real numbers, the temporal domain is abstractly specified by means of a set of first-order axioms defining the so-called $\mathcal{A}$-*models* [114].
The first-order neighborhood logic NL is quite expressive. In particular, it allows one to express the *chop* modality as follows:

$$\phi C\psi \triangleq \exists x, y(l = x + y) \wedge \Diamond_l\Diamond_r((l = x) \wedge \phi \wedge \Diamond_r((l = y) \wedge \psi)),$$

as well as any of the modalities corresponding to Allen's relations. Consequently, NL can virtually express all interesting properties of the underlying linear ordering, such as discreteness, density, etc.
Here we give an axiomatic system for NL, due to Barua, Roy, and Zhou [5], where the soundness and completeness proofs can be found. In what follows, the symbol $\Diamond$ stands either for $\Diamond_l$ and $\Diamond_r$, while $\overline{\Diamond}$ stands for $\Diamond_r$ (resp., $\Diamond_l$) when $\Diamond$ stands for $\Diamond_l$ (resp., $\Diamond_r$). The axiomatic system consists of the following axioms:

(A-NL1) $\Diamond\phi \rightarrow \phi$, where ϕ is a global formula;
(A-NL2) $l \geq 0$;
(A-NL3) $x \geq 0 \rightarrow \Diamond(l = x)$;
(A-NL4) $\Diamond(\phi \vee \psi) \rightarrow \Diamond\phi \vee \Diamond\psi$;
(A-NL5) $\Diamond\exists x\phi \rightarrow \exists x\Diamond\phi$;
(A-NL6) $\Diamond((l = x) \wedge \phi) \rightarrow \Box((l = x) \rightarrow \phi)$;
(A-NL7) $\Diamond\overline{\Diamond}\phi \rightarrow \Box\overline{\Diamond}\phi$;
(A-NL8) $(l = x) \rightarrow (\phi \leftrightarrow \overline{\Diamond}\Diamond((l = x) \wedge \phi)$;

(A-NL9) $((x \geq 0) \wedge (y \geq 0)) \rightarrow (\Diamond((l = x) \wedge \Diamond((l = y) \wedge \Diamond\phi)) \leftrightarrow \Diamond((l = x + y) \wedge \Diamond\phi))$,

plus the axioms for the domain $\mathbb{D}$ (axioms for $=, +, \leq$, and $-$), and the usual axioms for first-order logic. The same restrictions that have been made for the ITL concerning the instantiation of quantified formulas still apply here. The inference rules are, as usual, Modus Ponens, Necessitation, Generalization, and the following rule for Monotonicity:

$$\frac{\phi \rightarrow \psi}{\Diamond\phi \rightarrow \Diamond\psi}.$$

In [6], NL has been extended to a 'two-dimensional' version, called NL^2, where two modalities $\Diamond_u$ and $\Diamond_d$ have been added and interpreted as 'up' and 'down' neighbourhoods. NL^2 can be used to specify super-dense computations, taking vertical time as virtual time, and horizontal time as real time.

4.3 Duration calculi

Duration Calculus (DC for short) is an interval temporal logic endowed with the additional notion of **state**. Each state is denoted by means of a state expression, and it is characterized by a **duration**. The duration of a state is (the length of) the time period during which the system remains in the state. DC has been successfully applied to the specification and verification of real-time systems. For instance, it has been used to express the behaviour of communicating processes sharing a processor and to specify their scheduler, as well as to specify the requirements of a gas burner [107].
DC has originally been developed as an extension of Moszkowski's ITL, and thus denoted by DC/ITL. Since the seminal work by Zhou, Hoare, and Ravn [116], various meaningful fragments of DC/ITL have been isolated and analyzed. Recently, an alternative Duration Calculus, based on the logic NL, and thus denoted by DC/NL, has been proposed by Roy in [103]. As a matter of fact, most results for DC/ITL and its fragments transfer to DC/NL and its fragments. In the following we introduce the basic notions and we summarize the main results about DC/ITL. Further details can be found in [61].

4.3.1 The calculus DC/ITL. Zhou, Hoare, and Ravn's DC/ITL calculus is grounded on Moszkowski's ITL interpreted over the class of non-strict interval structures based on $\mathbb{R}$. Its only interval modality is *chop*. Its distinctive feature is the notion of state. States are represented by means of a new syntactic category, called **state expression**, which is defined as follows: the constants 0 and 1 are state expressions, a state variable X is a state expression, and, for any pair of state expression S and T, $\neg S$ and $S \vee T$ are state expressions (the other Boolean connectives are defined in the usual way). Furthermore, given a state expression S, the duration of S is denoted by $\int S$. DC/ITL terms are defined as in ITL, provided that temporal variables are replaced by state expressions. DC/ITL formulas are generated by the following abstract syntax:

$$\phi ::= p^n(r_1, \ldots, r_n) \mid \top \mid \neg\phi \mid \phi \vee \psi \mid \phi C \psi \mid \forall x \phi$$

where $r_1, \ldots, r_n$ are terms, p^n is a n-ary (global) predicate, C is the *chop* modality, and x is a global variable.

Any state (expression) S is associated with a total function $S : \mathbb{R} \mapsto \{0, 1\}$, which has a finite number of discontinuity points only. For any time point t, the state expression interpretation $\mathcal{I}$ is defined as follows:

- $\mathcal{I}[0](t) = 0$;
- $\mathcal{I}[1](t) = 1$;
- $\mathcal{I}[S](t) = S(t)$;
- $\mathcal{I}[\neg S](t) = 1 - \mathcal{I}[S](t)$;
- $\mathcal{I}[S \vee T](t) = 1$ if $\mathcal{I}[S](t) = 1$ or $\mathcal{I}[T](t) = 1$; 0 otherwise.

The semantics of a duration $\int S$ in a given (non-strict) model, with respect to an interval $[d_0, d_1]$, is the Riemann definite integral, that is, $\int_{d_0}^{d_1} \mathcal{I}[S](t)dt$. The semantics of the other syntactic constructs is given as in ITL case.

A number of useful abbreviations can be defined in DC/ITL. In particular, $\lceil S \rceil$ stands for: "S holds almost everywhere over a strict interval", and it is defined as follows: $\lceil S \rceil \triangleq (\int S = \int 1) \wedge \neg(\int 1 = \int 0)$; $\int 1$ is usually abbreviated by l, and it can be viewed as the length of the current interval; finally, $\lceil\ \rceil$, which holds over point-intervals, can be defined as $l = 0$.

The satisfiability problem for both first-order DC/ITL (full DC/ITL) and its fragment devoid of first-order quantification (Propositional DC/ITL) has been shown to be undecidable. First-order DC/ITL, provided with, at least, the functional symbol $+$ and the predicate symbol $=$, with the usual interpretation, has been completely axiomatized in [63]. The axiomatic system includes the following specific axioms:

(A-DC1) $\int 0 = 0$;
(A-DC2) $\int S \geq 0$;
(A-DC3) $\int S + \int T = \int (S \vee T) + \int (S \wedge T)$;
(A-DC4) $((\int S = x) C (\int S = y)) \leftrightarrow (\int S = x + y)$;
(A-DC5) $\int S = \int T$ provided that $S \leftrightarrow T$ holds in propositional logic

and the following inference rule (provided that $S_1 \ldots S_n$ are state expressions, and that $\bigvee_{i=1}^{n} \leftrightarrow 1$):

$$\frac{H(\lceil\ \rceil),\ H(\phi)\ \rightarrow\ H(\phi \vee \bigvee_{i=1}^{n} (\phi C \lceil S_i \rceil)}{H(\top)},$$

in conjunction with its inverse (obtained by exchanging the ordering of the formulas in every *chop*), where $H(\phi)$ represents the formula obtained from $H(X)$ by replacing every occurrence of X in H with ϕ.

Various interesting fragments of DC have been investigated by Zhou, Hansen, and Sestof in [115]. First, they consider the possibility of interpreting DC formulas over different classes of structures. In particular, the fragment of DC *interpreted over* $\mathbb{N}$ is the set of DC formulas interpreted over $\mathbb{R}$ evaluated with respect of $\mathbb{N}$-intervals, that is, intervals whose endpoints are in $\mathbb{N}$. The fragment of DC *interpreted over* $\mathbb{Q}$ is similarly defined. Then, the authors took into consideration some syntactic sub-fragments of the above calculi and studied the decidability/undecidability of their satisfiability problem. It turned out that the fragments of propositional DC whose formulas are built up from primitive formulas of the type $\lceil S \rceil$ only have a decidable satisfiability problem when interpreted over $\mathbb{N}$, $\mathbb{Q}$, and $\mathbb{R}$. By adding to the set of primitive formulas those of the form $l = k$, the problem remains decidable over $\mathbb{N}$, but it becomes undecidable over the other classes of structures. The same fragment at the first-order level is undecidable in all the considered cases. Finally, the fragment of propositional DC whose formulas are built up from primitive formulas of the type $\int S = \int T$ only is also undecidable.

As for the complexity of the satisfiability problem, in [95] Rabinovich reported a result by Sestoft (personal communication) stating that the satisfiability problem for the fragment of DC whose formulas are built up from primitive formulas of the type $\lceil S \rceil$ only, interpreted over $\mathbb{N}$, has a non-elementary complexity. Rabinovich showed that the satisfiability problem for the same fragment, interpreted over $\mathbb{R}$, also has a non-elementarily decidable, by providing a linear time reduction from the equivalence problem for star-free expressions to the validity problem for the considered fragment of DC.

In [26], Checuti-Sperandio and Fariñas del Cerro isolated another fragment of propositional DC by imposing suitable syntactic restrictions. Formulas of such a fragment are generated by the following abstract syntax:

$$\phi ::= \top \mid \bot \mid lPk \mid I = 0 \mid I = l \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi C \psi,$$

where k is a constant, $P \in \{<, \leq, =, \geq, >\}$, and I is $\int S$, for a given state S. The resulting logic is shown to be expressive enough to capture Allen's Interval Algebra. The authors developed a sound, complete, and terminating tableau system for the logic, thus showing that its satisfiability problem is decidable. The tableau system is a mixed procedure, combining standard tableau techniques with temporal constraint network resolution algorithms.

4.3.2 The calculus DC/NL. Finally, the classical DC and the first-order neighbourhood logic (NL) have been combined into the (clearly, undecidable) DC/NL which has been completely axiomatized by merging the axiomatic systems for DC and NL. The fragment of DC/NL obtained by restricting the formulas to be built up only from primitive formulas of the type $\lceil S \rceil$ has been proved to be decidable, while the extension of the latter with primitive formulas of the type $l = k$ is undecidable, as already mentioned.
Other variations of DC include the Propositional and First-Order Mean Value calculus, which have been studied by several authors including Pandya [93], and Zhou and Xiaoshan [117].

5 Summary and Additional References

In this survey paper we have attempted to give a picture of the extensive and rather diverse research done in the areas of interval temporal logics and duration calculi. We have focused our attention mainly on expressiveness issues, axiomatic systems, and (un)decidability results. Furthermore, we have presented a quite general tableau system for propositional interval logics.

To summarize, sound and complete axiomatic systems on propositional level are known for CDT with respect to certain classes of linear orderings, for HS with respect to the class of partial orderings with the linear intervals property, for the family of logics in $\mathcal{PNL}$ with respect to various classes of linear orderings, both in the strict and non-strict semantics, and for ITL and NL with respect to general semantics, while the problem of axiomatizing specific linear orderings is still largely unexplored. Also, sound and complete tableau systems have been developed for BCDT$^+$, which itself is, for the generality of that logic, actually a tableau method for an extensive variety of propositional interval logics, and for some local variants of ITL. The undecidability of the satisfiability/validity problem has been shown for HS, CDT, ITL, and NL, with respect to most classes of structures. As a matter of fact, rather weak subsystems of HS turn out to be (highly) undecidable for some classes of structures. On the other hand, decidable fragments have been obtained by imposing severe restrictions on the expressive power or the semantics of the logics, for instance, by imposing the locality projection principle. We point out again that, to the best of our knowledge, and axiomatic systems, decidability, etc. have not been explicitly addressed yet for the strict semantics variants of most of the existing interval logics (with the exceptions of PNL^{-} and its subsystems).

As any survey paper, this one cannot cover every topic of interest in the field. One relevant omission is that of many 'non-pure' interval logics. Interval-based temporal logics can be viewed as extensions of point-based temporal logics, where the notion of satisfaction at a state is replaced by the notion of satisfaction at an interval. From this point of view, the class of interval logics can be divided into two main classes: 'pure' interval logics, where the semantics is essentially interval-based, that is, propositional variables are evaluated with respect to a given interval, and 'non-pure' interval logics, where the semantics is essentially point-based and intervals are only auxiliary entities. In this survey paper, we have focused our attention on 'pure' interval logics, even though the interval logics with locality can be viewed, to a certain extent, as a particular case of non-pure interval logics. Important contributions in the area of 'non-pure' interval logics have been made by Dillon, Kutty, Moser, Melliar-Smith, and Ramakrishna (see [29,33,34,32,31], where Future Interval Logic and Graphical Interval Logic have been developed). Complexity results on 'non-pure' propositional interval logics have been obtained by Aaby and Narayana [1], while applications of these logics have been explored in Ramakrishna's PhD thesis [96]. There are other contributions that we have not discussed in our survey, such as, for instance, probabilistic interval logics [57]. Moreover, we have not discussed programming languages and related systems, based on interval logics [87,36]. Also, we have not discussed other types of deductive systems for interval logics such as sequent calculi and natural deduction, that have been proposed in the literature, e.g. in [110,99,97]. Finally, we have not given an account of the work on model-checking interval logics and duration calculi (see, e.g., [23,64]) which, undoubtedly, is an important and still largely unexplored topic.

Appendix

Here we sketch the proofs of some of the important results that we have stated in the survey.

Theorem 12 *The satisfiability problem for* PITL*-formulas interpreted discrete structure is not decidable.*

Proof.
This proof is actually an adaptation of the theorem by Chandra et al. [25] showing the undecidability of satisfiability for a propositional process logic. Given two context-free grammars G_1 and G_2 one can construct a PITL-formula that is satisfiable if and only if the intersection of the languages generated by the two grammars is nonempty. Since the latter problem is not decidable (see [66]), the claim follows. Without lose of generality, we assume that G_1 and G_2 do not contain $\varepsilon-$productions, that use 0 and 1 as the only terminal symbols, and that are in Greibach normal form (that is, the right-hand side of each production starts with a terminal symbols). For a given non-strict interval model $\mathbf{M}^{+}$ with domain $\mathbb{D}$ and valuation function V, and an interval $[d_0, d_1]$, we can define the **trace** $\sigma(p)$ of a propositional letter $p \in \mathcal{AP}$ in such a way that $\sigma_d(p) = 1$ if and only if $\mathbf{M}^{+}, [d, d] \Vdash p$ and $\sigma_d(p) = 0$ otherwise, for all $d \in \mathbb{D}$, $d_0 \leq d \leq d_1$. Similarly, $\sigma_{[d_0,d_1]}(p) = \sigma_{d_0}(p) \cdots \sigma_{d_1}(p)$. Suppose that a grammar

G is context-free, and consists of a list of m production sets $P = P_1, P_2, \ldots, P_m$, one for each non-terminal symbol $A_1, A_2 \ldots A_m$, that is, for all $1 \leq i \leq m$:

$$P_i : A_i \rightarrow P_{i,1} \mid P_{i,2} \mid \ldots \mid P_{i,|P_i|}$$

Let $L(G, A_i)$ be the language generated by G with A_i as starting symbol. We assume that $\mathcal{AP}$ contains the propositional letters $p, A_1, \ldots, A_m$ (p does not occur in G). We are going to build a $\mathrm{C}^{\pi z+}$ -formula $\phi(G, A_i)$ is such that $\mathbf{M}^+, [d_0, d_1] \Vdash \phi(G, A_i)$ if and only if $\sigma_{[d_0,d_1]} \in L(G, A_i)$. The formula is defined as follows:

- the translation of a terminal symbol 0 (resp., 1) is $\phi(G, 0) = \neg p \wedge \pi$ (resp., $\phi(G, 1) = p \wedge \pi$);
- the translation of a non-terminal symbol A_i is $\phi(G, A_i) = A_i$;
- the translation of a production string $P_{i,j} = V_1 V_2 \ldots V_{P_{i,j}}$ is $\phi(G, V_1\ V_2 \ldots V_{P_{i,j}}) = \phi(G, V_1)\ C(\bigcirc\pi)C\phi(G, V_2)C \ldots C(\bigcirc\pi)C\phi(G, V_{P_{i,j}})$;
- the translation of a production set P_i is $\phi(G, P_i) = (A_i \leftrightarrow [\phi(G, P_{i,1}) \vee \phi(G, P_{i,2}) \vee \ldots \vee \phi(G, P_{i,|P_i|})]$.

It easy to see that

$$\sigma_{[d_0,d_1]}(p) \in L(G, A_i) \text{ if and only if } \mathbf{M}^+, [d_0, d_1] \Vdash A_i \wedge \phi(G, P).$$

So, given two context-free grammars G_1, G_2 with disjoint sets of non-terminals and respective starting symbols A_1 and B_1, there exists a PITL-formula which is satisfiable if and only if the intersection of the languages generated by the two grammars is non-empty. □

Theorem 1 *The satisfiability problem for BE-formulas interpreted over dense structures is not decidable.*

Proof.
The idea of the proof is to represent a Turing Machine computation as an infinite sequence od IDs (**instantaneous description**, also called **configurations**). Each ID is a finite sequence of tape cells containing a unique tape symbol, and one of the cells has additional information representing the head position and the state of the machine. The core idea of the construction is the use of a proposition called *corr*, making possible to talk about consecutive IDs. Once this is done, the transition function δ of the Turing machine can be respected by examining a group of three cells in an ID and determining the value of the same three cells in the next ID. The claim is that the formula $computation \wedge comp - properties$ below, parameterized by a Turing Machine, is satisfiable if and only if it does not halt on a blank tape. We assume that the TM only writes symbols 0 and 1. Let Q be the set of states. We assume $\{0, 1, (q, 0), (q, 1), (q, B) \mid q \in Q)\} \in \mathcal{AP}$. Also, we use some other propositional symbols, like $cell, ID \ldots$. The formula $comp - properties$ enforces the properties that an interval $[d_0, d_1]$ must have when it satisfies the proposition $computation \in \mathcal{AP}$. So:

$$\begin{aligned} comp - properties = computation \rightarrow & \\ & not - contains(computation) \wedge \\ & seq(ID) \wedge seq(ID) - properties \wedge \\ & \langle B \rangle init - ID \wedge ID - properties \wedge \\ & cell - properties \wedge seq(cell) - properties \wedge \\ & corr - properties \wedge obeys - \delta. \end{aligned}$$

Now we explain the meaning of the above formulas. First,

$$not - contains(x) = \neg\langle B\rangle x \wedge \neg\langle D\rangle x \wedge \neg\langle E\rangle x$$

generically specifies that a *computation*−interval does not contain a sub-interval satisfying the same proposition *computation*.

$$\begin{aligned} seq(x) - properties = {} & [D](seq(x) \rightarrow \langle B\rangle x \wedge [E](x \rightarrow seq(x)) \\ & \wedge [D](x \rightarrow \neg seq(x))). \end{aligned}$$

The formula $ID - properties$ stands the properties that must hold for an ID:

$$ID - properties = [D](ID \rightarrow not - contains(ID) \wedge seq(cell) \wedge one - state)$$

where

$$one - state = \langle D\rangle state \wedge \neg\langle D\rangle(\langle B\rangle state \wedge \langle E\rangle state).$$

Some special ID are abbreviated, as:

$$init - ID = ID \wedge [D]((cell \rightarrow blank) \wedge (state \rightarrow init - state))$$

Below, some special cells are abbreviated:

$$\begin{aligned} cell - properties &= [D](cell \rightarrow not - contains(cell) \wedge unique - val) \\ unique - val &= \bigvee_{l \in \mathcal{AP}} l \wedge \bigwedge_{l,m \in \mathcal{AP}, l \neq m} (l \rightarrow \neg m) \\ init - state &= \bigvee_{(q_0,i) \in \mathcal{AP}} cell((q_0, i)) \\ cell(l) &= cell \wedge l \\ blank &= cell((B) \vee \bigvee_{(q,B) \in \mathcal{AP}} cell((q, B)). \end{aligned}$$

The proposition *corr* must be true at an interval if and only if it starts and ends with a cell, and these cells are corresponding cells in consecutive IDs. So:

$$\begin{aligned} corr - properties &= [D](cell - rule \wedge ID - rule \wedge corr - starts \wedge \\ &\quad corr - ends \wedge (corr \rightarrow not - contains(corr))) \\ cell - rule &= corr \rightarrow (\langle B\rangle cell \wedge \langle E\rangle cell) \\ corr - starts &= (\langle B\rangle cell \wedge \langle E\rangle corr) \rightarrow \langle B\rangle corr \\ corr - ends &= (\langle E\rangle cell \wedge \langle B\rangle corr) \rightarrow Ecorr \\ ID - rule &= (\langle B\rangle ID \wedge \langle E\rangle 2 - cell \wedge [E]\neg 3 - cell) \rightarrow corr. \end{aligned}$$

The meaning of the formulas $2-cell$ and $3-cell$ will be clear in the following. The transition function is respected by examining a group of three cells and determining the value of the middle state in the next ID.

$$3 - cell(x, y, z) = \langle B\rangle cell(x) \wedge \langle D\rangle cell(y) \wedge \langle E\rangle cell(z)$$
$$[D](cell \rightarrow cell(y))$$
$$3 - cell = \bigvee_{x,y,z} 3 - cell(x, y, z)$$
$$2 - cell = \langle B\rangle cell \wedge \langle E\rangle cell \wedge [D]\neg cell.$$

Finally, the δ function is codifies as follows:

$$obeys - \delta = \bigwedge_{i,j,k\in\mathcal{AP}} [D](\langle B\rangle corr \wedge \langle B\rangle 3 - cell(i, j, k) \rightarrow \langle E\rangle(cell(\delta(i, j, k)))).$$

As stated at the beginning, the formula $computation \wedge comp - properties$ is satisfied in a model $\mathbf{M}^+$ at an interval $[d_0, d_1]$ if and only if the corresponding Turing Machine does not halt on a blank tape. At the moment, this is true if, in the model, the interval $[d_0, d_1]$ is infinite (or, equivalently, over the ordinal ω^2). This shows that BE^+_ω is not decidable. Moreover, by adding a further level of detail, it is possible to show that the same property holds if $[d_0, d_1]$ is dense, showing the undecidability of BE interpreted over dense structures.

□

Theorem 17 *Every binary modal operator definable in* $FO_3[<](x_i, x_j)$ *has an equivalent in* CDT.

Proof.
Sketch. Soundness is straightforward. In order to prove completeness, one has to show that every $\text{FO}_3[<](x_i, x_j)$-formula (denoted by $f, g \ldots$) can be translated in an equivalent CDT-formula. Actually, since the truth of formulas in a CDT-model is only evaluated with respect to an interval, that is an ordered pair of points, while in classical logic there is no such a constraint, this problem is solved by giving two translations $f^>$ and $f^<$ of a given $f \in \text{FO}_3[<](x_i, x_j)$. It can be proved that the logic $\text{FO}_3[<](x_i, x_j)$ is equivalent to a simpler logic that we indicate by $L'(x_i, x_j)$, for which an inductive definition is possible: formulas of $L'(x_i, x_j)$ are $Q(x_i, x_j)$, $Q(x_j, x_i)$(where Q is a dyadic proposition symbol), $x_i = x_j$, $x_j = x_i$, $x_i < x_j$, $x_j < x_i$, boolean combinations of $L'(x_i, x_j)$-formulas, and $\exists x_k(f \wedge g)$, where $f, g \in L'(x_i, x_j)$. By exploiting this result, the translation ^{i,j} can be given by induction on the complexity of $L'(x_i, x_j)$ formulas.

- $(x_i = x_i)^{i,j} = \top$;
- $(x_i = x_j)^{i,j} = \pi$;
- $(x_j = x_j)^{i,j} = \top$;
- $(x_i < x_i)^{i,j} = \bot$;
- $(x_i < x_j)^{i,j} = \neg\pi$;
- $(x_j < x_i)^{i,j} = \bot$;
- $Q(x_i, x_i)^{i,j} = ((\pi \wedge q^>)C\neg\pi \vee (\pi \wedge q^>) \vee \neg\pi T(\pi \wedge q^>))$;
- $Q(x_i, x_j)^{i,j} = q^>$;
- $Q(x_j, x_i)^{i,i} = q^<$;
- $Q(x_j, x_j)^{i,i} = \neg\pi C(\pi \wedge q^>) \vee (\pi \wedge q^>) \vee \neg\pi D(\pi \wedge q^>)$;
- $(\neg\phi)^{i,j} = \neg\phi^{i,j}$;

- $(\phi \wedge \psi)^{i,j} = \phi^{i,j} \wedge \psi^{i,j}$;
- $(\exists x_k(\phi(x_i, x_j) \wedge \psi(x_k, x_j)))^{i,j} = \phi^{k,i} D \psi^{i,k} \vee \phi^{i,k} C \psi^{k,j} \vee \psi^{j,k} T \phi^{i,k}$.

The symbols $q^>$ and $q^<$ corresponds to the propositional letter in the adapted CDT-models. It is possible to prove that for every CDT-formula ϕ there exists a corresponding equivalent pair of formulas $\phi^>$ and $\phi^<$ interpreted over adapted CDT-models. The main result is that, for every $L'(x_i, x_j)$-formula f, there exists a pair of CDT-formulas $\phi^>$ and $\phi^<$, obtained by the above translation, which is satisfiable if and only if f is satisfiable, and this concludes the proof. □

Theorem 7 *The satisfiability problem for* HS*-formulas interpreted over the class of all structures with an infinitely ascending sequence is not decidable.*

Proof.
Sketch. This proof can actually be viewed as a generalization of the one of the theorem 1. The idea is to construct formulas that essentially encode the computation of a Turing Machine. More precisely, the idea is to have an HS-formula which is satisfiable if and only if a Turing Machine with a given program and started on a black tape never halts. Since the non-halting problem is co-r.e.-hard, this makes the satisfiability problem co-r.e.-hard, and the validity problem r.e.-hard. The construction proceeds as follows. A Turing Machine is fixed, and it is assumed that it can write only symbols 0 and 1. Let Q be the set of states, with q_0 being the unique starting state, and q_f the unique halting state. It is assumed that $\mathcal{AP}$ contains the propositional letters $\{0, 1, \#, *, (q, 0), (q, 1), (q, B) \mid q \in Q\}$. The computation started on a black tape in state q_0 is encoded as a sequence if IDs separated by pairs of special symbols $(*)$. Each ID consists of a sequence of cells, each one of them contains one of the elements of the language, e.g. $(q, 0)$ or $(q, 1)$. The construction of the various parts of the formula is quite similar to the one in Theorem 1. □

Copy of: **V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. Journal of Universal Computer Science, 9(9):1137–1167, 2003.**
Included with permission from the Executive Editor of JUCS.

Abstract. Logics for time intervals provide a natural framework for dealing with time in various areas of computer science and artificial intelligence, such as planning, natural language processing, temporal databases, and formal specification. In this paper we focus our attention on propositional interval temporal logics with temporal modalities for neighboring intervals over linear orders. We study the class of propositional neighborhood logics ($\mathcal{PNL}$) over two natural semantics, respectively admitting and excluding point-intervals. First, we introduce interval neighborhood frames and we provide representation theorems for them; then, we develop complete axiomatic systems and semantic tableaux for logics in $\mathcal{PNL}$.

1 Introduction

Logics for time intervals provide a natural framework for dealing with time in various areas of computer science and artificial intelligence, such as planning and natural language processing, where reasoning about time intervals rather than time points is far more natural and closer to common sense (differences and similarities between point-based and interval-based temporal logics are systematically analyzed in [7]). Various interval temporal logics have been proposed in the literature. The most important propositional ones are Halpern and Shoham's Modal Logic of Time Intervals (HS) [59] and Venema's CDT logic [110], while relevant first-order interval logics are Zhou and Hansen's Neighborhood Logic (NL) [114] and Moszkowski's Interval Temporal Logic (ITL) [84]. (In [51] we survey the main developments, results, and open problems on interval temporal logics and duration calculi.)
HS features four basic operators: $\langle B\rangle$ (*begin*) and $\langle E\rangle$ (*end*), and their transposes $\langle\overline{B}\rangle$ and $\langle\overline{E}\rangle$. Given a formula φ and an interval $[d_0,d_1]$, $\langle B\rangle\varphi$ holds at $[d_0,d_1]$ if φ holds at $[d_0,d_2]$, for some $d_2<d_1$, and $\langle E\rangle\varphi$ holds at $[d_0,d_1]$ if φ holds at $[d_2,d_1]$, for some $d_2>d_0$. All other temporal operators corresponding to Allen's relations can be defined by means of the basic ones. In particular, it is possible to define the (strict) *after* operator $\langle A\rangle$ (resp., its transpose $\langle\overline{A}\rangle$) such that $\langle A\rangle\varphi$ (resp., $\langle\overline{A}\rangle\varphi$) holds at $[d_0,d_1]$ if φ holds at $[d_1,d_2]$ (resp., $[d_2,d_0]$) for some $d_2>d_1$ (resp., $d_2<d_0$), and the sub-interval operator $\langle D\rangle$ such that $\langle D\rangle\phi$ holds at a given interval $[d_0,d_1]$ if ϕ holds at a proper sub-interval $[d_2,d_3]$ of $[d_0,d_1]$. Complete axiomatic systems for HS with respect to several classes of structures are given in [109], while the undecidability of HS over various linear orders has been proved in [59] by encoding the halting problem in it. Venema's CDT has three binary operators, namely, C (*chop*), D, and T, which correspond to ternary interval relations occurring when an extra point is added in one of the three possible distinct positions with respect to the two endpoints of the current interval (*between, before*, and *after*), and a modal constant π which holds over an interval $[d_0,d_1]$ if $d_0=d_1$. Axiomatic systems for CDT can be found in [110]. Since HS can be embedded into CDT, the undecidability of the latter follows from that of the former. Furthermore, in [74] Lodaya shows that the fragment of HS that only contains $\langle B\rangle$ and $\langle E\rangle$, interpreted over dense linear orders, is already undecidable. Since both $\langle B\rangle$ and $\langle E\rangle$ can be easily defined in terms of C and π, it immediately follows that a logic only provided with C and π is undecidable as well.
Zhou and Hansen's NL features the two 'expanding' modalities $\Diamond_r$ and $\Diamond_l$ and a special symbol l denoting the length of the current interval. Given a formula φ and an interval $[d_0,d_1]$, $\Diamond_r\varphi$ holds at $[d_0,d_1]$ if φ holds at $[d_1,d_2]$, for some $d_2\geq d_1$; $\Diamond_l\varphi$ holds at $[d_0,d_1]$

if φ holds at $[d_2, d_0]$, for some $d_2 \leq d_0$; and the valuation of l over $[d_0, d_1]$ is $d_1 - d_0$. Some properties, applications, and extensions of NL are given in [6,103], while a complete axiomatic system can be found in [5]. NL undecidability can be easily proved by embedding HS in it. Moszkowski's ITL is a first-order interval logic providing two modalities, namely $\bigcirc$ (*next*) and C. In ITL an interval is defined as a finite or infinite sequence of states. Given two formulas φ, ψ and an interval $s_0, \ldots, s_n$, $\bigcirc\varphi$ holds over $s_0, \ldots, s_n$ if φ holds over $s_1, \ldots, s_n$, while $\varphi C \psi$ holds over $s_0, \ldots, s_n$ if there exists i, with $0 \leq i \leq n$, such that φ holds over $s_0, \ldots, s_i$ and ψ holds over $s_i, \ldots, s_n$. Studies of axiomatic systems and completeness for fragments and extensions of ITL include [37,56]. ITL has been proved to be undecidable even at the propositional level by a reduction from the problem of testing the emptiness of the intersection of two grammars in Greibach form [84]. As a matter of fact, the results given in [74] prove the undecidability of (a variant of) ITL with the C and the modal constant ϕ for point-intervals, interpreted over dense linear ordering. A decidable fragment of propositional ITL with quantification over propositional variables has been obtained by imposing a suitable *locality* constraint [84]. Such a constraint states that each propositional variable is true over an interval if (and only if) it is true at its first state. This allows one to collapse all the intervals starting at the same state into the single interval consisting of the first state only. By exploiting such a constraint, decidability of Local ITL can be easily proved by embedding it into Quantified Propositional Linear Temporal Logic.

In order to model duration properties of real-time systems, both NL and ITL have been extended with a notion of 'state variable' that represents an instantaneous observation of the system behavior. In particular, the Duration Calculus (DC) extends ITL by adding *temporal variables* (also called state expressions) as integrals of state variables [114,116,62]. Temporal variables make it possible to represent the duration of intervals as well as numerical constants. As an example [107], the specification of the behavior of a gas burner can include conditions as the following one: "for any period of 30 seconds the gas may leak, that is, flow and not burn, only once and for 4 seconds at most". Such a condition is expressed by the DC formula: $l > 30 \vee ((\int(\neg Gas \vee \ Flame) = l); (\int(Gas \wedge \neg Flame) = l \wedge l \leq 4); (\int(\neg Gas \vee Flame) = l))$, where *Gas* (the gas is flowing) and *Flame* (the gas is burning) are two state variables. In [115] Zhou et al. show that DC is undecidable, the main source of undecidability being the fact that state changes in real-time systems can occur at any time point.

In this paper we study propositional interval neighborhood temporal logics, the family of which we denote by $\mathcal{PNL}$. Logics in $\mathcal{PNL}$ can express meaningful timing properties, without being excessively expressive to an extent easily leading to high undecidability, a typical phenomenon for interval logics. They feature two modalities which correspond to Allen's *meet* and *met by* relations [2], intuitively capturing a *right* neighboring interval and a *left* neighboring interval. There are two natural semantics for interval logics interpreted over linearly ordered domains, namely a *non-strict* one, which includes intervals with coincident endpoints (*point-intervals*), and a *strict* one, which excludes them and was already studied in [4,73]. To make it easier to distinguish between the two semantics from the syntax, the modal operators $\Diamond_r$ and $\Diamond_l$ are used in the case of non-strict propositional neighborhood logics, generically denoted by $\mathcal{PNL}^+$, while for the strict ones, denoted by $\mathcal{PNL}^-$, $\Diamond_r$ and $\Diamond_l$ are replaced by $\langle A \rangle$ and its transpose $\langle \overline{A} \rangle$, respectively. While the logics in $\mathcal{PNL}^+$are built on the propositional fragment of NL, those in $\mathcal{PNL}^-$can be viewed as based on the $A\overline{A}$-fragment of HS. In fact, the semantics of HS admits point-intervals and hence, according to our classification, it is non-strict. However, the modalities $\langle A\rangle$ and $\langle\overline{A}\rangle$ only refer to strict intervals, and thus the semantics of the fragment $A\overline{A}$ can be considered essentially strict.

The main contributions of the paper are: (i) representation theorems for strict and non-strict interval neighborhood structures; (ii) complete axiomatic systems for logics in $\mathcal{PNL}$; (iii) complete semantic tableaux for $\mathcal{PNL}$ logics. Unlike classical logic and most modal and temporal logics, where the first-order axiomatic systems are obtained by extending their propositional fragments with relevant axioms for the quantifiers, the first-order NL was axiomatized first, without its propositional fragment having been identified. It now turns out that the latter was hidden into the originally introduced first-order axiomatic system, the propositional axioms of which, taken alone, are substantially incomplete. In particular, a curious feature of NL is that while it can be finitely axiomatized, its propositional fragment involves an infinite axiom scheme. The strict analogue, however, is a finitely axiomatized subsystem of the latter. As for the tableaux, there is no a straightforward way of adapting existing tableaux for point-based propositional and first-order temporal logics to interval temporal logics [112,39]. We develop an original tableau method for $\mathcal{PNL}$ logics which combines features of classical first-order tableau and point-based temporal tableaux. (For a detailed account of the existing tableau methods see [28].)

There are very few tableau methods for time interval logics and duration calculi in the literature. In [11], Bowman and Thompson consider an extension of Local ITL, which, besides the chop operator C, contains a projection operator *proj* and the modal constant π. They introduce a normal form for the formulas of the resulting logic that allows them to exploit a classical tableau method, devoid of any mechanism for constraint label management. In [26], Chetcuti-Sperandio and Fariñas del Cerro identify a decidable fragment of DC, which is expressive enough to model the above-given condition on the behavior of a gas burner, that imposes no restriction on state expressions, but encompasses a proper subset of DC operators, namely, $\wedge$, $\vee$, and C. The tableau construction for the resulting logic combines application of the rules of classical tableaux with that of a suitable constraint resolution algorithm and it essentially depends on the assumption of *bounded* variability of the state variables. Finally, tableau systems for the propositional and first-order Linear Temporal Logic (LTL), which employ a mechanism for labeling formulas with temporal constraints somewhat similar to ours, are given in [104] and [24], respectively. The main differences between these tableau methods and ours are: (i) they are specifically designed to deal with integer time structures (i.e., linear and discrete) while ours is quite generic; (ii) LTL is essentially point-based, and intervals only play a secondary role in it (viz., a formula is true on an interval if and only if it is true at every point in it), while in our systems intervals are primary semantic objects on which the truth definitions are entirely based; (iii) the closedness of a tableau is defined in terms of unsatisfiability of the associated set of temporal constraints, while in our system it is entirely syntactic.

The rest of the paper is organized as follows. In Section 2 we introduce interval neighborhood frames and structures; then, in Section 3 we provide representation theorems for both strict and non-strict semantics. In Section 4 we give syntax and semantics of $\mathcal{PNL}$, and in Section 5 we define various logics in this class. In Section 6 we briefly discuss the expressive power of $\mathcal{PNL}$ and we give some examples of its use. Sections 7 and 8 are devoted to the axiomatic systems and respective completeness theorems for both semantics. In Section 9 we develop semantic tableaux for $\mathcal{PNL}$ logics, and prove their soundness and completeness. In the last section we provide an assessment of the work done and we briefly discuss our ongoing research on $\mathcal{PNL}$.

2 Interval Neighborhood Frames and Structures

In this section, we introduce the basic notions of interval neighborhood frame and structure.

Definition 5. *A* ***neighborhood frame*** *is a triple* $\mathbf{F} = \langle \mathbb{I}, R, L\rangle$ *where* $\mathbb{I}$ *is a non-empty set and* R, L *are binary relations on* $\mathbb{I}$.

For every sequence $S_1, ..., S_k \in \{R, L\}$, we denote the composition of the relations $S_1, ..., S_k$ by $S_1...S_k$. Also, we put:

$\mathbf{B_F} = \{w \in \mathbb{I} \mid \text{there is no } v \in \mathbb{I} \text{ such that } wLv\}$,
$\mathbf{B^2_F} = \{w \in \mathbb{I} \mid \text{there are no } u, v \in \mathbb{I}, \text{ with } u \neq v, \text{ such that } wLv \text{ and } wLu\}$,
$\mathbf{E_F} = \{w \in \mathbb{I} \mid \text{there is no } v \in \mathbb{I} \text{ such that } wRv\}$, and
$\mathbf{E^2_F} = \{w \in \mathbb{I} \mid \text{there are no } u, v \in \mathbb{I}, \text{ with } u \neq v, \text{ such that } wRv \text{ and } wRu\}$.

Consider the following conditions:

(NF1) R and L are mutually inverse;
(NF2) $\forall x\forall y(\exists z(xLz\wedge zRy)\rightarrow \forall z(xLz\rightarrow zRy))$ and
$\forall x\forall y(\exists z(xRz\wedge zLy)\rightarrow\forall z(xRz \rightarrow zLy))$;
(NF3) $RL \subseteq LRR \cup LLR \cup E$ on $\mathbb{I} - \mathbf{B^2_F}$ and $LR \subseteq RLL \cup RRL \cup E$ on $\mathbb{I} - \mathbf{E^2_F}$, where E is the equality, that is,
$\forall x\forall y(\exists z\exists u(xLz \wedge zLu)\wedge\exists z(xRz\wedge zLy) \rightarrow x = y\vee$
$\exists w\exists z((xLw\wedge wRz\wedge zRy)\vee (xLw\wedge wLz\wedge zRy)))$
and
$\forall x\forall y(\exists z\exists u(xRz \wedge zRu)\wedge \exists z(xLz \wedge zRy) \rightarrow x = y\vee$
$\exists w \exists z((xRw \wedge wLz \wedge zLy)\vee(xRw\wedge wRz\wedge zLy)))$;
(NF4) $RRR \subseteq RR$, i.e. $\forall w\forall x\forall y\forall z(wRx\wedge xRy\wedge yRz \rightarrow \exists u(wRu\wedge uRz))$.

Definition 6. *An* ***interval neighborhood frame*** *is a neighborhood frame* $\mathbf{F} = \langle \mathbb{I}, R, L\rangle$ *satisfying the conditions NF1,...,NF4.*

Note that, assuming **NF1, NF4** is equivalent to

$$\forall w\forall x\forall y\forall z(wLx \wedge xLy \wedge yLz \rightarrow \exists u(wLu \wedge uLz)).$$

Definition 7. *An interval neighborhood frame* $\mathbf{F} = \langle \mathbb{I}, R, L\rangle$ *is said to be:*

- ***strict,*** *if the relation* LRR *is irreflexive, and* ***non-strict*** *if the relation* LRR *is reflexive (note that 'not strict' does not imply 'non-strict');*
- ***open,*** *if it satisfies the condition* $\forall x(\exists y(xLy)\wedge\exists y(xRy))$;
- ***rich,*** *if it satisfies the condition* $\forall x(\exists y(xRy\wedge yRy) \wedge \exists y(xLy\wedge yLy))$;
- ***normal,*** *if it satisfies the condition* $\forall x\forall y(\forall z(zRx \leftrightarrow zRy)\wedge\forall z(zLx \leftrightarrow zLy)\rightarrow x = y)$;
- ***tight,*** *if it satisfies the condition* $\forall x\forall y((xRRy \wedge yRRx)\rightarrow x = y)$;
- ***weakly left-connected*** *(resp.,* ***weakly right-connected****) if the relation* $LR \cup LRR \cup LLR$ *(resp.,* $RL \cup RRL \cup RLL$*) is an equivalence relation on* $\mathbb{I} - \mathbf{B_F}$ *(resp.,* $\mathbb{I} - \mathbf{E_F}$*);* ***left-connected*** *(resp.,* ***right-connected****) if that relation is the universal relation on* $\mathbb{I} - \mathbf{B_F}$ *(resp.,* $\mathbb{I} - \mathbf{E_F}$*);*

– ***weakly connected*** *if each of the relations* $LR \cup LRR \cup LLR$ *and* $RL \cup RRL \cup RLL$ *is an equivalence relation on* $\mathbb{I}$*;* ***connected****, if each of these relations is the universal relation on* $\mathbb{I}$*.*

Now, consider the following definitions:

(NF5) NF2 implies $LRL \subseteq L$ and $RLR \subseteq R$, that is,
$\forall x \forall y((xLRLy \to xLy) \land (xRLRy \to xRy))$;
(NF6) assuming NF2, normality implies
$\forall x \forall y(\exists z(zRx \land zRy) \land \exists z(zLx \land zLy) \to x = y)$, that is,
$\forall x \forall y(xLRy \land xRLy \to x = y)$.
Assuming also openness, normality becomes equivalent to that condition;
(NF7) in every non-strict interval neighborhood frame, $RR = RRR$ and $LL = LLL$;
(NF8) every rich interval neighborhood frame is non-strict and open;
(NF9) every non-strict interval neighborhood frame is weakly connected. Every strict interval neighborhood frame is weakly left- and right-connected;
(SNF) in every strict interval neighborhood frame each of L, R, LLR, RRL, and RLL is irreflexive, too;
(NNF) an interval neighborhood frame is non-strict iff either of $LRR \cup LLR$ and $RLL \cup RRL$ is an equivalence relation on $\mathbb{I}$.

Proposition 2. *NF5, NF6, NF7, NF8, SNF, and NNF are consequences of the definitions.*

Eventually, we are interested in *concrete* interval neighborhood structures.

Definition 8. *If* $\langle \mathbb{D}, < \rangle$ *is a linearly ordered domain, an* ***interval*** *in* $\mathbb{D}$ *is a pair* $[d_0, d_1]$ *such that* $d_0, d_1 \in \mathbb{D}$ *and* $d_0 \leq d_1$*.* $[d_0, d_1]$ *is a* ***strict interval*** *if* $d_0 < d_1$*, while it is a* ***point interval*** *if* $d_0 = d_1$*.*

Definition 9. *A* ***non-strict interval neighborhood structure*** *is a neighborhood frame* $\langle \mathbb{I}(\mathbb{D})^+, R, L \rangle$*, where* $\mathbb{I}(\mathbb{D})^+$ *is the set of all intervals over some linear ordering* $\langle \mathbb{D}, < \rangle$ *and* R, L *are mutually inverse binary relations over* $\mathbb{I}(\mathbb{D})^+$ *such that* vRw *holds if and only if* w *is a* ***right neighbor*** *of* v*, i.e.* $v = [d_0, d_1]$ *and* $w = [d_1, d_2]$ *for some* $d_0, d_1, d_2 \in \mathbb{D}$*. Then* v *is said to be a* ***left neighbor*** *of* w*. The substructure of the interval neighborhood structure* $\langle \mathbb{I}(\mathbb{D})^+, R, L \rangle$ *containing only the strict intervals will be called* ***strict interval neighborhood structure****, denoted by* $\langle \mathbb{I}(\mathbb{D})^-, R, L \rangle$*.*

Proposition 3. *Every strict (resp., non-strict) interval neighborhood structure is a strict (resp., non-strict) interval neighborhood frame.*

Proof. Straightforward. ☐

If a linear order $\langle \mathbb{D}, < \rangle$ has a particular property (i.e. it is dense, discrete, unbounded, etc.), we say that the interval neighborhood structure based on it has that property.

3 Representation Theorems for Interval Neighborhood Frames

In this section, we provide representation theorems for both strict and non-strict semantics (as for the strict case, it must be noted that similar representation results can be found in [73]).

Theorem 22 (Non-strict Representation Theorem). *If* $\mathbf{F}$ *is a tight, rich, connected, and normal interval neighborhood frame, then* $\mathbf{F}$ *is isomorphic to a non-strict interval neighborhood structure.*

Proof. Let $\mathbf{F} = \langle \mathbb{I}, R, L \rangle$ be a tight, rich, connected, and normal interval neighborhood frame. We construct an underlying linear ordering for $\mathbf{F}$ and then we show that $\mathbf{F}$ is isomorphic to the non-strict interval neighborhood structure over that ordering.
Let $\mathbf{P}(\mathbb{I}) = \{u \in \mathbb{I} | uRu\}$. Note that $\mathbf{P}(\mathbb{I})$ is non-empty and uLu for every $u \in \mathbf{P}(\mathbb{I})$. We will show that for every $u, v \in \mathbf{P}(\mathbb{I})$,

$$uLRv \text{ iff } u = v.$$

Indeed, $uLuRu$, i.e. $uLRu$. Conversely, let $uLRv$. Note that, by NF5, LR is an equivalence relation on $\mathbf{P}(\mathbb{I})$. Furthermore, if $uLRv$ then $uRuLRv$, i.e. $uRLRv$, so uRv, hence $uRLv$ and so, likewise, uLv. Now, for every w, vRw implies $uRLRw$, hence uRw. Likewise, uRw implies vRw. Analogously, uLw implies vLw and vice versa. Then, by normality, $u = v$. From this, it follows that for every $w \in \mathbb{I}$ there is a unique $v \in \mathbf{P}(\mathbb{I})$, hereafter denoted by $\mathbf{b}(w)$, such that wLv. Likewise, there is a unique $v \in \mathbf{P}(\mathbb{I})$, hereafter denoted by $\mathbf{e}(w)$, such that wRv. We now define a relation $\preceq$ on $\mathbf{P}(\mathbb{I})$ as follows:

$$u \preceq v \text{ iff } uRRv.$$

The relation $\preceq$ is a linear ordering on $\mathbf{P}(\mathbb{I})$: reflexivity is obvious, transitivity follows from NF7 and NF8, and anti-symmetry follows from tightness. As for the linearity: for any $u, v \in \mathbf{P}(\mathbb{I})$, $uLRRv$ or $uLLRv$ since $LRR \cup LLR$ is the universal relation on $\mathbb{I}$. Suppose $uLRRv$. Then $uRuLRRv$, i.e., $uRLRRv$, hence $uRRv$, i.e., $u \preceq v$. Likewise, if $uLLRv$ then $uLLv$, hence $vRRu$, i.e., $v \preceq u$. Note that for every $w \in \mathbb{I}$, $\mathbf{b}(w)RwR\mathbf{e}(w)$, hence $\mathbf{b}(w) \preceq \mathbf{e}(w)$. Now, we define a mapping μ from $\mathbb{I}$ to the non-strict interval neighborhood structure $\langle \mathbb{I}^+(\mathbf{P}(\mathbb{I})), \mathbf{L}, \mathbf{R} \rangle$ over $\langle \mathbf{P}(\mathbb{I}), \preceq \rangle$ as follows:

$$\mu(w) = (\mathbf{b}(w), \mathbf{e}(w)).$$

1. μ is an injection. If $\mu(w_1) = \mu(w_2)$, then let $\mathbf{b}(w_1) = \mathbf{b}(w_2) = b$ and $\mathbf{e}(w_1) = \mathbf{e}(w_2) = e$. Then, for every $x \in \mathbb{I}$, w_1Rx implies $w_2R\mathbf{e}(w_2)(= \mathbf{e}(w_1))Lw_1Rx$, i.e., w_2RLRx, hence w_2Rx. Likewise, w_2Rx implies w_1Rx. Analogously, w_1Lx iff w_2Lx. Then, by normality, $w_1 = w_2$.
2. μ is onto. If $u, v \in \mathbf{P}(\mathbb{I})$ and $u \preceq v$, then $uRRv$, i.e., $uRwRv$ for some $w \in \mathbb{I}$ and hence $\mu(w) = (u, v)$.
3. μ is an isomorphism. If w_1Rw_2, then $\mathbf{e}(w_1)R\mathbf{e}(w_1)Lw_1Rw_2$, that is, $\mathbf{e}(w_1)\ RLRw_2$. Hence $\mathbf{e}(w_1)Rw_2$, and thus $\mathbf{e}(w_1) = \mathbf{b}(w_2)$ by uniqueness of $\mathbf{b}(w_2)$. It follows that $\mu(w_1)\mathbf{R}\mu(w_2)$. Conversely, if $\mu(w_1)\mathbf{R}\mu(w_2)$, then $w_1R\mathbf{e}(w_1)L\ \mathbf{e}(w_1)\ (= \mathbf{b}(w_2))Rw_2$, i.e., $w_1\ RLRw_2$, and hence w_1Rw_2. Likewise, w_1Lw_2 iff $\mu(w_1)\ \mathbf{L}\ \mu(w_2)$.

This completes the proof. □

Theorem 23 (Strict Representation Theorem). *We have that:*

1. *Every weakly connected, strict and normal interval neighborhood frame is isomorphic to a strict interval neighborhood structure;*
2. *Every connected, open, strict and normal interval neighborhood frame is isomorphic to a strict unbounded interval neighborhood structure.*

Proof. We prove 2 (the proof can be easily modified for 1). Let $\mathbf{F}^- = \langle \mathbb{I}, R, L\rangle$ be a connected, open, strict, and normal interval neighborhood frame. We construct an underlying point-based linear ordering and we show that $\mathbf{F}^-$ is isomorphic to the strict unbounded interval neighborhood structure over that ordering.

First, for every $w \in \mathbb{I}$, we define $[w]_b = \{v \in \mathbb{I} \mid wLRv\}$ and $[w]_e = \{v \in \mathbb{I} \mid wRLv\}$. By NF5, we have that $LRL \subseteq L$ and $RLR \subseteq R$. Hence, the relations LR and RL are equivalence relations in $\mathbb{I}$, and thus the sets $P_b = \{[w]_b \mid w \in \mathbb{I}\}$ and $P_e = \{[w]_e \mid w \in \mathbb{I}\}$ are partitions of $\mathbb{I}$. Now, we define the mapping $\theta : P_e \mapsto P_b$ as follows:

$$\theta([w]_e) = [v]_b \text{ where } wRv.$$

First, note that the definition is correct: if $[w_1]_e = [w_2]_e$, $[v_1]_b = [v_2]_b$, and w_1Rv_1 then $w_2RLRLRv_2$. By NF5, we obtain w_2RLRv_2 and thus w_2Rv_2 by NF5 again. Then, θ is a function: if wRv_1 and wRv_2 then v_1LRv_2, i.e., $[v_1]_b = [v_2]_b$; also, if wRv and $w_1 \in [w]_e$, then w_1RLw. Hence w_1RLRv, and thus w_1Rv. Furthermore, θ is a bijection between P_e and P_b. Indeed, if $\theta([w_1]_e) = \theta([w_2]_e) = [v]_b$, then w_1Rv and w_2Rv, and hence w_1RLw_2, i.e., $[w_1]_e = [w_2]_e$. The surjectivity immediately follows from the definition of P_b. From now on, we will identify P_e with P_b via θ and we will only deal with P_b. We define a relation $<$ on P_b as follows:

$$[w]_b < [v]_b \text{ iff } wLRRv.$$

Correctness of the definition: if $[w_1]_b = [w_2]_b$, $[v_1]_b = [v_2]_b$, and $w_1LRR\,v_1$, then w_2LRw_1LRR v_1LRv_2, i.e., $w_2(LRL)R(RLR)v_2$, and thus w_2LRRv_2 by NF5. Now we show that the relation $<$ is a strict linear ordering on P_b:

1. Irreflexivity holds because $\mathbf{F}^-$ is strict.
2. Transitivity: let $w_1LRRw_2LRRw_3$, i.e., $w_1LR(RLR)Rw_3$. Hence, we have that w_1 $LRRR$ w_3 by NF5, and thus w_1 $LRRw_3$ by NF4.
3. Linearity: we have to show that for every $[w]_b, [v]_b \in P_b$, $[w]_b < [v]_b$ or $[w]_b = [v]_b$ or $[v]_b < [w]_b$, i.e., $wLRRv$ or $wLRv$ or $vLRRw$, that is, $wLLRv$, which is precisely the connectedness condition on $\mathbf{F}^-$.

Note that $\langle P_b, <\rangle$ is open: for every $[w]_b \in P_b$ there exists $v \in \mathbb{I}$ such that vRw and there exists $u \in \mathbb{I}$ such that vLu. Hence, $vLRRw$, i.e., $[v]_b < [w]_b$. Likewise, there exists $[v]_b$ such that $[w]_b < [v]_b$. It remains to show that the strict interval structure on $\langle P_b, <\rangle$ is isomorphic to $\mathbf{F}^-$. The isomorphism is given by the mapping $\mu : \mathbf{F}^- \mapsto \mathbb{I}(P_b)^-$ determined by

$$\mu(w) = ([w]_b, \theta([w]_e)).$$

Let $\theta([w]_e) = [v]_b$ where wRv. We have that $wLRRv$, and thus $[w]_b < [v]_b$. Hence, μ associates intervals from $\mathbb{I}(P_b)^-$ with every $w \in \mathbf{F}^-$. Now, if $[w_1]_b = [w_2]_b$ and $\theta([w_1]_e) = \theta([w_2]_e)$, then w_1LRw_2, and w_1Rv_1 and w_2Rv_2, for v_1, v_2 such that $[v_1]_b = [v_2]_b$ and thus v_1LRv_2. Hence w_1RLRLw_2, and thus w_1RLw_2 by NF5. From w_1LRw_2 and w_1RLw_2, it follows that $w_1 = w_2$ by NF6, that is, NF2 plus normality. Finally, for every interval $([w]_b, [v]_b)$ in $\mathbb{I}(P_b)^-$, we have $[w]_b < [v]_b$, i.e., $wLRRv$, and thus $wLRu$ and uRv for some $u \in \mathbf{F}^-$. Then $[u]_b = [w]_b$ and $\theta([u]_e) = [v]_b$, i.e., $([w]_b, [v]_b) = \mu(u)$. Thus, μ is an isomorphism and the proof is completed. □

4 Propositional Neighborhood Logics: Syntax and Semantics

The language $\mathbf{L}^+$ for the class $\mathcal{PNL}^+$ of **non-strict propositional neighborhood logics** contains a set of propositional variables $\mathcal{AP}$, the propositional logical connectives $\neg$ and $\rightarrow$, and the modalities $\Box_r$ and $\Box_l$, the dual operators of which will be denoted by $\Diamond_r$ and $\Diamond_l$, respectively. The remaining classical propositional connectives, as well as the logical constants $\top$ (true) and $\bot$ (false), can be considered as abbreviations.
The **formulas** of $\mathcal{PNL}^+$, denoted by $\phi, \psi, \ldots$, are recursively defined as follows:

$$\phi = p \mid \neg\phi \mid \phi\wedge\psi \mid \Box_r\phi \mid \Box_l\phi.$$

The language $\mathbf{L}^-$ for the class $\mathcal{PNL}^-$ of **strict propositional neighborhood logics** differs from $\mathbf{L}^+$ only in the notation for the modalities, now denoted by $[A]$ and $[\overline{A}]$, with dual operators $\langle A\rangle$ and $\langle\overline{A}\rangle$, respectively. The **formulas** of $\mathcal{PNL}^-$ are defined as follows:

$$\phi = p \mid \neg\phi \mid \phi\wedge\psi \mid [A]\phi \mid [\overline{A}]\phi.$$

We use different notations for the modalities in $\mathbf{L}^+$ and $\mathbf{L}^-$ only to reflect their historical links and to make it easier to distinguish between the two semantics from the syntax. Clearly, there is a straightforward translation between the two languages.

The semantics of a propositional neighborhood logic is given in **non-strict** or **strict models**, respectively based on non-strict and strict interval neighborhood frames and equipped with a **valuation function** for the propositional variables. Valuation functions are defined as $V : \mathbb{I} \mapsto 2^{\mathcal{AP}}$ in such a way that, for any $p \in \mathcal{AP}$ and $w \in \mathbb{I}$, if $p \in V(w)$, then p is **true** over w, otherwise it is **false**. **Satisfiability** at an interval $w \in \mathbb{I}$ in a non-strict (resp., strict) model $\mathbf{M}$ is defined by induction on the structure of the formulas:

1. $\mathbf{M}, w \Vdash p$ iff $p \in V(w)$, for all $p \in \mathcal{AP}$;
2. $\mathbf{M}, w \Vdash \neg\psi$ iff it is not the case that $\mathbf{M}, w \Vdash \psi$;
3. $\mathbf{M}, w \Vdash \phi\wedge\psi$ iff $\mathbf{M}, w \Vdash \phi$ and $\mathbf{M}, w \Vdash \psi$;
4. $\mathbf{M}, w \Vdash \Box_l\phi$ (resp., $[\overline{A}]\phi$) iff for every interval v such that wLv we have $\mathbf{M}, v \Vdash \phi$;
5. $\mathbf{M}, w \Vdash \Box_r\phi$ (resp., $[A]\phi$) iff for every interval v such that wRv we have $\mathbf{M}, v \Vdash \phi$.

We will also take into consideration the extension of logics in $\mathcal{PNL}^+$ with the modal constant π:

6. $\mathbf{M}^+, w \Vdash \pi$ iff w is a point-interval.

Finally, the relevant notion of **p-morphism** between (non-strict or strict) models for $\mathcal{PNL}$ is defined in a standard way, and it satisfies the usual truth-preservation property well-known from modal/temporal logics (see e.g. [7]).

5 Some Propositional Neighborhood Logics

The logics of the (valid formulas in the) classes of all non-strict, respectively strict, interval neighborhood structures will be denoted by PNL$^+$, respectively PNL$^-$. Besides the valid formulas in the class of all interval neighborhood structures, we will be interested in some natural subclasses of non-strict or strict interval structures:

– the **unbounded** linear orderings, denoted by **u**;

- the **dense** linear orderings (between every two different points there is a point), denoted by **de**;
- the **discrete** linear orderings (every point having a successor, respectively, a predecessor, has an immediate one), denoted by **di**;
- the **Dedekind complete** linear orderings (where every non-empty and bounded above set of points has a least upper bound), denoted by **c**;
- the **unbounded and dense** linear orderings, denoted by **ude**;
- the **unbounded and discrete** linear orderings, denoted by **udi**;
- the **unbounded and Dedekind complete** linear orderings, denoted by **uc**.

The logics of these classes will be denoted accordingly: in the class $\mathcal{PNL}^+$ by $\mathrm{PNL}^{\lambda+}$ and in the class $\mathcal{PNL}^-$ by $\mathrm{PNL}^{\lambda-}$, where $\lambda \in \{\mathbf{u}, \mathbf{de}, \mathbf{di}, \mathbf{c}, \mathbf{ude}, \mathbf{udi}, \mathbf{uc}\}$. For example, the logic PNL^{udi+} is the logic of the valid formulas in all non-strict unbounded and discrete neighborhood structures. Furthermore, the logic PNL^+ (resp., $\mathrm{PNL}^{\lambda+}$) endowed with π will be denoted by $\mathrm{PNL}^{\pi+}$ (resp., $\mathrm{PNL}^{\lambda\pi+}$).

Consider the following formulas:

(A-SNF^{ur}) $[A]p \rightarrow \langle A\rangle p$ (or, equivalently, $\langle A\rangle\top$);

(A-SNF^{der}) $(\langle A\rangle\langle A\rangle p \rightarrow \langle A\rangle\langle A\rangle\langle A\rangle p) \wedge (\langle A\rangle[A]p \rightarrow \langle A\rangle\langle A\rangle[A]p)$;

(A-SNF^{aux}) $\langle\overline{A}\rangle\top \rightarrow \langle\overline{A}\rangle\langle\overline{A}\rangle\top$;

(A-SNF^{dir}) $([A]\bot \rightarrow [\overline{A}]([A][A]\bot \vee \langle A\rangle(\langle A\rangle\top \wedge [A][A]\bot))) \wedge$
$((\langle A\rangle\top \wedge [A](p \wedge [\overline{A}]\neg p \wedge [A]p)) \rightarrow [\overline{A}][\overline{A}]\langle A\rangle(\langle A\rangle\neg p \wedge [A][A]p))$;

(A-SNF$^{\mathbf{c}}$) $\langle A\rangle\langle A\rangle[\overline{A}]p \wedge \langle A\rangle[A]\ \neg[\overline{A}]p \rightarrow \langle A\rangle(\langle A\rangle[\overline{A}]\ [\overline{A}]p \wedge [A]\langle A\rangle\neg[\overline{A}]p)$.

Proposition 4. *In the strict semantics:*

1. *The class of all unbounded structures is defined by the formula A-SNF^{ur} and its inverse* ***A-SNF^{ul}*** *(let* ***A-SNF^{u}*** *be A-SNF^{ur} $\wedge$ A-SNF^{ul}).*
2. *The class of all dense structures, extended with the 2-element linear ordering*[4]*, is defined by the formula A-SNF^{der} and its inverse* ***A-SNF^{del}*** *or, alternatively, the formula A-SNF^{aux} (let* ***A-SNF^{de}*** *be A-SNF^{der} $\wedge$ A-SNF^{del}).*
3. *The class of all discrete structures is defined by the formula A-SNF^{dir} and its inverse* ***A-SNF^{dil}*** *(let* ***A-SNF^{di}*** *be A-SNF^{dir} $\wedge$ A-SNF^{dil}).*
4. *The class of all Dedekind complete structures is defined by the formula A-SNF$^{\mathbf{c}}$.*
5. *The class of all unbounded and dense structures is defined by the formulas A-SNF^{ur}, A-SNF^{der}, and their inverses A-SNF^{ul}, A-SNF^{del}.*
6. *The class of all unbounded and discrete structures is defined by the formulas A-SNF^{ur}, A-SNF^{dir}, and their inverses A-SNF^{ul}, A-SNF^{dil}.*
7. *The class of all unbounded and Dedekind complete structures is defined by the formulas A-SNF^{ur} and its inverse A-SNF^{ul}, and A-SNF$^{\mathbf{c}}$.*

Proof. Sketch:

1. Straightforward.

[4] The 2-element linear ordering cannot be separated in the language of PNL^-.

2. The formula A-SNF^{der} says that every interval with a left neighbor can be split into two sub-intervals. In addition, A-SNF^{aux} guarantees that if there are at least 2 intervals (i.e., at least 3 points), then the left-most interval, if there is one, can be split into two sub-intervals, too.
3. The formula A-SNF^{dir} (resp. A-SNF^{dil}) says that every point which has a successor (resp. predecessor) has an immediate one.
4. The formula A-SNF^{c} says that every non-empty and bounded above set of points has a least upper bound. □

Proposition 5. *The above-defined logics satisfy the following relations:*

1. *For every $\lambda_1, \lambda_2 \in \{u, de, di, c, ude, udi, uc\}$, $\mathrm{PNL}^{\lambda_1 -} \subsetneq \mathrm{PNL}^{\lambda_2 -}$ if and only if the class of linear orders characterized by the condition λ_2 is strictly contained in the class of linear orders characterized by the condition λ_1;*
2. *$\mathrm{PNL}^{ude-} \subsetneq \mathrm{PNL}^{+}$, where the inclusion is in terms of the obvious translation between the two languages.*
3. *$\mathrm{PNL}^{+} = \mathrm{PNL}^{u+} = \mathrm{PNL}^{de+} = \mathrm{PNL}^{ude+} = \mathrm{PNL}^{di+} = \mathrm{PNL}^{udi+}$.*

Proof. Sketch:

1. First, $\mathrm{PNL}^{-} \subsetneq \mathrm{PNL}^{u-}$ because the formula A-SNF^{u} $\in \mathrm{PNL}^{u-} - \mathrm{PNL}^{-}$. Likewise, $\mathrm{PNL}^{de-} \subsetneq \mathrm{PNL}^{ude-}$. $\mathrm{PNL}^{-} \subsetneq \mathrm{PNL}^{de-}$ because the formula A-SNF^{de} $\in \mathrm{PNL}^{de-} - \mathrm{PNL}^{-}$, since it is valid in every strict and dense neighborhood structure, but e.g. not in the one based on $\mathbb{Z}$. Likewise, $\mathrm{PNL}^{u-} \subsetneq \mathrm{PNL}^{ude-}$. $\mathrm{PNL}^{-} \subsetneq \mathrm{PNL}^{di-}$ because the formula A-SNF^{di} $\in \mathrm{PNL}^{di-} - \mathrm{PNL}^{-}$, since it is valid in every strict and discrete neighborhood structure, but not in the one based on $\mathbb{Q}$. Likewise, $\mathrm{PNL}^{u-} \subsetneq \mathrm{PNL}^{udi-}$. $\mathrm{PNL}^{-} \subsetneq \mathrm{PNL}^{c-}$ because the formula A-SNF^{c} $\in \mathrm{PNL}^{c-} - \mathrm{PNL}^{-}$, since it is valid in every Dedekind-complete strict neighborhood structure, but not in the one based on $\mathbb{Q}$. Finally, we have that $\mathrm{PNL}^{di-} \subsetneq \mathrm{PNL}^{udi-}$, $\mathrm{PNL}^{u-} \subsetneq \mathrm{PNL}^{uc-}$, and $\mathrm{PNL}^{c-} \subsetneq \mathrm{PNL}^{uc-}$.
2. Every PNL^{+}-formula satisfiable in a model $\mathbf{M}^{+}$ over non-strict neighborhood structure is satisfiable in a dense and unbounded strict one. Indeed, replacing every point in $\mathbf{M}^{+}$ by a copy of $\mathbb{Q}$ produces a dense and unbounded strict model $\mathbf{M}^{-*}$ such that $\mathbf{M}^{+}$ is a p-morphic copy of $\mathbf{M}^{-*}$.
3. Essentially the same construction works for the equalities $\mathrm{PNL}^{+} = \mathrm{PNL}^{u+} = \mathrm{PNL}^{de+} = \mathrm{PNL}^{ude+}$, but now we take the non-strict version of $\mathbf{M}^{*}$. For the equality $\mathrm{PNL}^{+} = \mathrm{PNL}^{udi+}$, we can similarly replace every point in $\mathbf{M}$ by a copy of $\mathbb{Z}$, and thus produce an unbounded and discrete non-strict model which maps p-morphically onto $\mathbf{M}$. □

It is worth noting that the logic PNL^{udi-} does not yet characterize the interval structure of the integers, because the formula

$$\langle A\rangle p \wedge [A](p \rightarrow \langle A\rangle p) \wedge [A][A](p \rightarrow \langle A\rangle p) \rightarrow [A]\langle A\rangle\langle A\rangle p$$

is valid in the integers, but not in PNL^{udi-} since it fails in a PNL^{udi-}-model based on $\mathbb{Z} + \mathbb{Z}$.

The above proposition shows that there is a collapse of the expressiveness in the non-strict semantics, while the strict one is at least as expressive as the point-based temporal logic over linear orders. The situation is graphically depicted in Figure 1, where $\lambda \in \{u, de, di, ude, udi\}$.

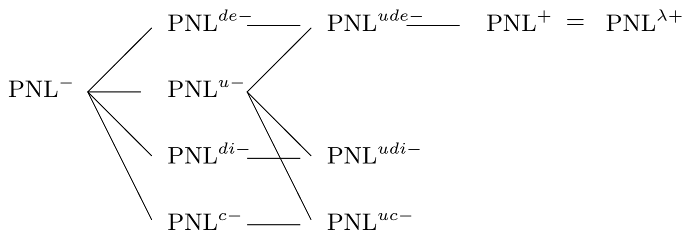


Fig. 1. Relative expressive power of $\mathcal{PNL}$ logics.

6 Expressing Timing Properties in $\mathcal{PNL}$

Here we give some simple examples of properties that can be expressed in $\mathcal{PNL}$. First of all, note that PNL$^-$, besides distinguishing among different properties of the underlying linear order, is powerful enough to express the **difference** operator:

$$\overline{D}(q) \equiv [\overline{A}][\overline{A}][A]q \wedge [\overline{A}][A][A]q \wedge [A][A][\overline{A}]q \wedge [A][\overline{A}][\overline{A}]q,$$

and consequently to simulate **nominals**: $n(q) \equiv q \wedge [\neq](\neg q)$, that is, to express the fact that q holds in the current interval and nowhere else. Therefore, every universal property of strict interval structures can be expressed in PNL$^-$.

The following more practical examples are borrowed from typical application domains in Artificial Intelligence. As a first example, consider the case of a robot that, in order to accomplish a given goal, must pick a finite set of objects $a_1, a_2, \ldots, a_n$ in whatever order. Moreover, assume that the robot cannot pick up more than one object at a time. Such a scenario can be modeled as follows. Let the propositional variable p_{a_i}, with $1 \leq i \leq n$, denote the action "the robot is picking up the objects a_i" and the propositional variable $h_{a_{i_1},\ldots,a_{i_k}}$, with $a_{i_j} \in \{a_1, \ldots, a_n\}$ and $1 \leq k \leq n$, denote the state "the robot holds the objects $a_{i_1}, \ldots, a_{i_k}$". The constraint that picking up and holding each object is a necessary pre-condition of any situation in which the robot simultaneously holds all objects can be expressed in PNL$^+$ as follows:

$$h_{a_1,\ldots,a_n} \rightarrow \Diamond_l \Diamond_l (p_{a_1} \wedge \Diamond_r h_{a_1}) \wedge \ldots \wedge \Diamond_l \Diamond_l (p_{a_n} \wedge \Diamond_r h_{a_n})$$

Note that such a formulation does not constrain "picking up" actions to be instantaneous. However, such a condition can be easily expressed in PNL$^{\pi+}$:

$$h_{a_1,\ldots,a_n} \rightarrow \Diamond_l \Diamond_l (p_{a_1} \wedge \pi \wedge \Diamond_r h_{a_1}) \wedge \ldots \wedge \Diamond_l \Diamond_l (p_{a_n} \wedge \pi \wedge \Diamond_r h_{a_n}).$$

As another example, we note that both PNL$^+$ and PNL$^-$ allows one to define an interval version of the **until** operator by means of the formulas

$$\phi \Diamond_u \psi \equiv \Diamond_r (\phi \wedge \Diamond_r \psi) \quad \text{and} \quad \phi \langle U \rangle \psi \equiv \langle A \rangle (\phi \wedge \langle A \rangle \psi),$$

respectively. Such an operator can be used to express conditions of the form "The flight from Milano to Johannesburg initiates a period of time during which the traveler is in Johannesburg" as follows:

$$\mathit{Milano\text{-}to\text{-}Johannesburg} \langle U \rangle \mathit{Stay\text{-}in\text{-}Johannesburg}.$$

Moreover, in $\mathrm{PNL}^{\pi+}$ one can express the constraint that "the non-instantaneous period of time during which the light is on is initiated (resp. terminated) by an instantaneous action of switch on (resp. switch off)" as follows:

$$Switch\text{-}On \wedge \pi \wedge ((Light\text{-}On \wedge \neg\pi)\Diamond_u(Switch\text{-}Off \wedge \pi)).$$

As a matter of fact, the proposed interval version of the until operator suffers from some limitations. In particular, to obtain a decomposable version of it we should force homogeneity either implicitly (via the assumption of the homogeneity principle [3]) or explicitly (by means of sub-interval operators). Besides the well-known fields of planning and natural language processing, successful applications of interval temporal logics can be found in the areas of digital system design and verification [84] and of model validation phase support [94]. As for Moszkowski's ITL, the logic $\mathrm{PNL}^{di\pi+}$ can be exploited to express various interesting statements about digital systems. As an example, one can constrain "the output q of a device to strictly follow the input p" (being p and q two non-instantaneous states of the device) as follows:

$$(\neg\pi \wedge p) \rightarrow (\neg\pi\Diamond_u(\neg\pi \wedge q)).$$

Other useful statements about digital systems can be captured by exploiting the difference operator. As for the model validation task, interval temporal logics have been used to keep significantly low the number of states to be checked in HSTS Planner, a model-based planning system of the Remote Agent autonomous system architecture [94].

7 Axiomatic Systems for $\mathcal{PNL}^+$

7.1 An Axiomatic System for PNL^+

We propose the following axioms for PNL^+, where the inverse of a formula is obtained by interchanging $\Box_r$ and $\Box_l$:

(A-NT) enough propositional tautologies;

(A-NK) the K axioms for $\Box_r$ and $\Box_l$;

(A-NNF0) $\Box_r p \rightarrow \Diamond_r p$, and its inverse;

(A-NNF1) $p \rightarrow \Box_r \Diamond_l p$, and its inverse;

(A-NNF2) $\Diamond_r \Diamond_l p \rightarrow \Box_r \Diamond_l p$, and its inverse;

(A-NNF3) $\Box_r \Diamond_l p \rightarrow \Diamond_l \Diamond_r \Diamond_r p \vee \Diamond_l \Diamond_l \Diamond_r p$, and its inverse;

(A-NNF4) $\Diamond_r \Diamond_r \Diamond_r p \rightarrow \Diamond_r \Diamond_r p$, and its inverse;

(A-NNF$_\infty$) $\Box_r q \wedge \Diamond_r p_1 \wedge \ldots \wedge \Diamond_r p_n \rightarrow \Diamond_r(\Box_r q \wedge \Diamond_r p_1 \wedge \ldots \wedge \Diamond_r p_n)$, and its inverse, for each $n \geq 1$.

The rules of inference are, as usual, Modus Ponens, Uniform Substitution, and $\Box_r$ and $\Box_l$ Generalization.

Proposition 6. *A neighborhood frame $\mathbf{F}^+ = \langle \mathbb{I}, R, L \rangle$ is an interval neighborhood frame if and only if the axioms A-NNF1,...,A-NNF4 are valid in $\mathbf{F}^+$.*

Proof. It is simple to check that the axioms A-NNF1, ..., A-NNF4 modally define the semantic conditions NF1-NF4 in the non-strict semantics. □

We show that the given axiomatic system for PNL^+ is sound and complete.

Lemma 1. *The following formulas and their inverses are derivable in* PNL^+*:*

1. $\Diamond_r p \rightarrow \Box_r \Box_l \Diamond_r p$;
2. $\Diamond_r \Diamond_l \Diamond_r p \rightarrow \Diamond_r p$;
3. $\Diamond_l \Diamond_r p \rightarrow \Diamond_r \Diamond_l \Diamond_l p \vee \Diamond_r \Diamond_r \Diamond_l p$.

Proof. For 1, use A-NNF1 and A-NNF2. For 2, observe that $\mathrm{PNL}^+ \vdash \Diamond_r \Diamond_l \Diamond_r p \rightarrow \Diamond_r \Box_l \Diamond_r p$ by Axiom A-NNF2 (and Axiom A-NNF0), hence $\mathrm{PNL}^+ \vdash \Diamond_r \Diamond_l \Diamond_r p \rightarrow \Diamond_r p$ by Axiom A-NNF1. Finally, 3 follows from A-NNF2 and A-NNF3. ☐

Lemma 2. *Let* $\mathbf{M}^{+^*} = \langle \mathbb{I}^*, R^*, L^*, V^* \rangle$ *be any generated sub-model of the canonical model for* PNL^+ *and let* $w \in \mathbb{I}^*$*. Then there is* $w_b \in \mathbb{I}^*$ *such that* $\{\phi \mid \Box_l \phi \in w\} \cup \{\Diamond_l \psi \mid \Diamond_l \psi \in w\} \cup \{\Box_l \xi \mid \Box_l \xi \in w\} \subseteq w_b$*, and* $w_e \in \mathbb{I}^*$ *such that* $\{\phi \mid \Box_r \phi \in w\} \cup \{\Diamond_r \psi \mid \Diamond_r \psi \in w\} \cup \{\Box_r \xi \mid \Box_r \xi \in w\} \subseteq w_e$*.*

Proof. It suffices to show that the set $\Gamma = \{\phi \mid \Box_l \phi \in w\} \cup \{\Diamond_l \psi \mid \Diamond_l \psi \in w\} \cup \{\Box_l \xi \mid \Box_l \xi \in w\}$ is PNL^+-consistent. Suppose otherwise. Then for some ϕ such that $\Box_l \phi \in w$, $\Box_l \xi \in w$, and $\{\Diamond_l \psi_1, \dots \Diamond_l \psi_n\} \subseteq w$, the set $\{\phi, \Diamond_l \psi_1, \dots \Diamond_l \psi_n, \Box_l \xi\}$ is PNL^+-inconsistent, i.e., $\mathrm{PNL}^+ \vdash \phi \rightarrow \neg(\Box_l \xi \wedge \Diamond_l \psi_1 \wedge \dots \wedge \Diamond_l \psi_n)$. Hence $\mathrm{PNL}^+ \vdash \Box_l \phi \rightarrow \Box_l \neg(\Box_l \xi \wedge \Diamond_l \psi_1 \wedge \dots \wedge \Diamond_l \psi_n)$. Thus $\Box_l \neg(\Box_l \xi \wedge \Diamond_l \psi_1 \wedge \dots \wedge \Diamond_l \psi_n) \in w$, i.e., $\neg \Diamond_r(\Box_l \xi \wedge \Diamond_l \psi_1 \wedge \dots \Diamond_l \psi_n) \in w$. On the other hand, $\Diamond_l(\Box_l \xi \wedge \Diamond_l \psi_1 \wedge \dots \wedge \Diamond_l \psi_n) \in w$ by A-NF$_\infty$, which is a contradiction. Thus, Γ is contained in a maximal PNL^+-consistent set w_b in $\mathbb{I}^*$. The existence of w_e is proved likewise. ☐

Theorem 24 (Soundness and Completeness). PNL^+ *is (sound and) complete for the class of all non-strict interval neighborhood structures.*

Proof. Soundness is straightforward. Note that the truth of most axioms, including the axiom scheme A-NNF$_\infty$, hinges on the inclusion of point intervals.
For the completeness, we take any PNL^+-consistent formula ϕ. It is satisfied at the root w of some generated sub-model $\mathbf{M}^+$ of the canonical model for PNL^+. Regarding that generated sub-model as a first-order structure of the language with $=$, R, L, and unary predicates corresponding to the atomic propositions occurring in ϕ, we take (using the Downwards Löwenheim-Skolem theorem) a countable elementary substructure $\mathbf{M}^{+^*}$ of $\mathbf{M}^+$ containing w. Let $\mathbf{M}^{+^*} = \langle \mathbf{F}^{+^*}, V^* \rangle$, where $\mathbf{F}^{+^*} = \langle \mathbb{I}^*, R^*, L^* \rangle$. The elements of $\mathbb{I}^*$ will henceforth be called 'intervals'. Note that $\mathbf{M}^{+^*}, w \Vdash \phi$ since truth of an interval formula at a given interval of a given PNL^+-model is a first-order property. Furthermore, Lemma 2 implies the truth of the first-order formulas $\forall x(\exists y(xLy \wedge \forall t(xLt \leftrightarrow yLt)))$ and $\forall x(\exists z(xRz \wedge \forall t(xRt \leftrightarrow zRt)))$ in $\mathbf{M}^+$ and hence in $\mathbf{M}^{+^*}$. Thus, with every interval v, $\mathbf{M}^{+^*}$ contains intervals v_b and v_e satisfying the conditions of Lemma 2. Note also that the 'point intervals' in $\mathbf{M}^{+^*}$ are distinguished by being both R^* -reflexive and L^*-reflexive. (In fact, one reflexivity implies the other since R^* and L^* are mutually inverse.)
Now, let w_b and w_e be as in Lemma 2. We are going to build step-by-step an interval neighborhood structure, mapping p-morphically over $\mathbf{F}^{+^*}$. We will inductively define a chain of interval neighborhood structures $\mathbf{F}_0^+ \subseteq \dots \mathbf{F}_n^+ \subseteq \dots$, where $\mathbf{F}_n^+ = \langle \mathbb{I}(\mathbb{D})_n, R_n, L_n \rangle$, and a sequence of mappings $f_n : \mathbf{F}_n^+ \mapsto \mathbf{F}^{+^*}$, for $n = 0, 1, 2, \dots$, satisfying the conditions: (i) $yR_n z \rightarrow f_n(y) R^* f_n(z)$, and (ii) $yL_n z \rightarrow f_n(y) L^* f_n(z)$, as follows. Let $\mathbb{D}_0 = \{d_0, d_1\}$, with $d_0 < d_1$. R_0 and L_0 are standard right neighbor and left neighbor relations on $\mathbb{I}(\mathbb{D})_0^+$.

$f_0([d_0, d_1]) = w$, $f_0([d_0, d_0]) = w_b$, and $f_0([d_1, d_1]) = w_e$. Clearly, the function f_0 preserves the right and left neighbor relations.

Suppose now that $\mathbf{F}_n^+$ and f_n are defined and satisfy the conditions (i) and (ii). Let $\mathbb{D}_n = \{d_0, \ldots, d_n\}$, where $d_0 < \ldots < d_n$. In general, f_n is not a p-morphism from $\mathbf{F}_n^+$ to $\mathbf{F}^{+*}$ because there are *p-morphism defects* in $\mathbf{F}_n^+$ which we will have to repair during the construction, viz.: the image under f_n of an interval $[d_k, d_m]$ in $\mathbf{F}_n^+$ has a right neighbor (resp., a left neighbor) v in $\mathbf{F}^{+*}$, which is 'missing' in $\mathbf{F}_n^+$, i.e., v is not an f_n-image of any interval from $\mathbb{I}(\mathbb{D})_n^+$, related likewise to $[d_k, d_m]$. Let all possible defects, i.e., pairs of neighboring intervals from $\mathbf{F}^{+*}$ (which are countably many since $\mathbf{F}^{+*}$ is countable), each repeated countably many times, be listed in a sequence $\mathcal{D} = \{\delta_n\}_{n<\omega}$, and let δ be the first one in the sequence, which has not been dealt with yet, and which occurs in $\mathbf{F}_n^+$. We are going to expand $\mathbf{F}_n^+$ to $\mathbf{F}_{n+1}^+$ in such a way that the defect δ will be fixed.

Suppose that δ relates the (image of the) interval $[d_k, d_m]$ from $\mathbf{F}_n^+$ and, say, a right neighbor v of $f_n([d_k, d_m])$ in $\mathbf{F}^{+*}$, which is not an image of any interval from $\mathbf{F}_n^+$. (In particular, that means that $f_n([d_k, d_m]) \neq v$.) We then extend $\mathbf{F}_n^+$ to $\mathbf{F}_{n+1}^+$ with a new point d_h and f_n to f_{n+1} so that $f_{n+1}([d_m, d_h]) = v$. We must still find an appropriate place of d_h in the linear ordering $\mathbb{D}_n$ and define f_{n+1} over all other intervals with an endpoint d_h in a way which preserves the neighborhood relations. Note that $f_n([d_k, d_m])R^*R^*L^*v$, hence $f_n([d_0, d_m])R^*R^*R^*L^*v$, and so $f_n([d_0, d_m])R^*R^*L^*v$ by axiom A-NNF4. Let d_{m+i} be the greatest element of $\mathbf{F}_n^+$ such that $f_n([d_0, d_{m+i}])\ R^*R^*L^*v$. Then, for each $j = 0, \ldots, m+i$, $f_n([d_0, d_j])\ R^*R^*L^*\ f_n([d_0, x_{m+i}])$, so $f_n([d_0, d_j])\ R^*R^*L^*R^*R^*L^*v$, and hence $f_n([d_0, d_j])\ R^*R^*L^*v$ by Lemma 1 (part 2) and axiom A-NF4. Therefore, for each $j = 0, \ldots, m+i$, there is $w_j \in \mathbf{F}^{+*}$ such that $f_n([d_0, d_j])R^*w_j$ and $w_jR^*L^*v$.

We now place d_h between d_{m+i} and d_{m+i+1} (if $m+1 \leq n$, otherwise we place d_h to the right of d_n) and extend f_n over all new intervals as follows. First, we put $f_{n+1}([d_m, d_h]) = v$. Then, for each $j = 1, \ldots, m+i$, $j \neq m$, we define $f_{n+1}([d_j, d_h]) = w_j$. For $j > m+i$, it is not the case that $f_n([d_m, d_j])R^*R^*L^*v$ (otherwise, $f_n([d_0, d_j])R^*R^*L^*v$). On the other hand, $f_n([d_m, d_j])L^*R^*v$ because $f_n([d_m, d_j])L^*f_n([d_k, d_m])$ and $f_n([d_k, d_m])R^*v$ by assumption. Then, by Lemma 1 (part 3), $f_n([d_m, d_j])R^*L^*L^*v$. Therefore, there exists $w_j \in \mathbf{F}^{+*}$ such that $f_n([d_m, d_j])R^*L^*w_j$ and w_jL^*v. We define $f_{n+1}([d_h, d_j]) = w_j$. Finally, choose $v_e \in \mathbf{F}^{+*}$ satisfying the condition of Lemma 2 and put $f_{n+1}([d_h, d_h]) = v_e$. It is straightforward to check that conditions (i) and (ii) still hold for $\mathbf{F}_{n+1}^+$. For example, if $d_j < d_h < d_l$, then $[d_j, d_h]R_{n+1}[d_h, d_h]$, and thus $f_{n+1}([d_j, d_h])R^*\ L^*v$ and $f_{n+1}([d_h, d_l])L^*v$. Hence $f_{n+1}([d_j, d_h])\ R^*L^*R^*f_{n+1}\ ([d_h, d_l])$, and therefore $f_{n+1}([d_j, d_h])\ R^*f_{n+1}\ ([d_h, d_l])$. This completes the inductive procedure.

Now, we define $\mathbb{D}_\omega = \bigcup_{n<\omega} \mathbb{D}_n$, $L_\omega = \bigcup_{n<\omega} L_n$, $R_\omega = \bigcup_{n<\omega} R_n$, $f_\omega = \bigcup_{n<\omega} f_n$ and $\mathbf{F}_\omega^+ = \langle \mathbb{I}(\mathbb{D})_\omega^+, R_\omega, L_\omega \rangle$. Finally, we define a valuation V_ω in $\mathbf{F}_\omega^+$ according to V^* in $\mathbf{F}^{+*}$, viz. for all $p \in \mathcal{AP}$, $V_\omega(p) = \{i \in \mathbb{I}(\mathbb{D})_\omega^+ \mid f_\omega(i) \in V^*(p)\}$. Let $\mathbf{M}_\omega^+ = \langle \mathbf{F}_\omega^+, V_\omega \rangle$. Then $f_\omega : \mathbf{M}_\omega^+ \mapsto \mathbf{M}^{+*}$ is a surjective p-morphism, hence $\mathbf{M}_\omega^+, [d_0, d_1] \Vdash \phi$. $\square$

7.2 An Axiomatic System for $\mathrm{PNL}^{\pi+}$

We extend the axiomatic system for PNL^+ to $\mathrm{PNL}^{\pi+}$ by adding the following axioms:

(A-π1**)** $\Diamond_l \pi \wedge \Diamond_r \pi$;

(A-π2**)** $\Diamond_r(\pi \wedge p) \rightarrow \Box_r(\pi \rightarrow p)$ and its inverse $\Diamond_l(\pi \wedge p) \rightarrow \Box_l(\pi \rightarrow p)$;

(A-π3**)** $\Diamond_r p \wedge \Box_r q \rightarrow \Diamond_r(\pi \wedge \Diamond_r p \wedge \Box_r q)$ and its inverse

$\Diamond_l p \wedge \Box_l q \rightarrow \Diamond_l(\pi \wedge \Diamond_l p \wedge \Box_l q)$.

By induction on n, one can show that all formulas $\Diamond_r(\pi \wedge p_1) \wedge ... \wedge \Diamond_r(\pi \wedge p_n) \rightarrow \Diamond_r(\pi \wedge p_1 \wedge ... \wedge p_n)$ and their inverses are derivable in PNL$^{\pi+}$, and thus that $\Box_r q \wedge \Diamond_r p_1 \wedge ... \wedge \Diamond_r p_n \rightarrow \Diamond_r(\pi \wedge \Box_r q \wedge \Diamond_r p_1 \wedge ... \wedge \Diamond_r p_n)$ is derivable as well. Therefore, the infinite scheme A-NNF$_\infty$ becomes derivable, hence redundant, in PNL$^{\pi+}$.

The completeness proof for PNL$^+$ is readily adaptable to PNL$^{\pi+}$.

8 Axiomatic Systems for $\mathcal{PNL}^-$

8.1 An Axiomatic System for PNL$^-$

Except for the scheme A-NF$_\infty$, which is no longer valid, the axioms for PNL$^-$ are very similar to those for PNL$^+$ (accordingly modified to reflect the fact that point-intervals are now excluded), where $\Diamond_r, \Diamond_l$ are replaced by $\langle A\rangle, \langle\overline{A}\rangle$, and $\Box_r, \Box_l$ accordingly by $[A], [\overline{A}]$. We propose the following system for PNL$^-$:

(A-ST) enough propositional tautologies;

(A-SK) the K axioms for $[A]$ and $[\overline{A}]$;

(A-SNF1) $p \rightarrow [A]\langle\overline{A}\rangle p$ and its inverse;

(A-SNF2) $\langle A\rangle\langle\overline{A}\rangle p \rightarrow [A]\langle\overline{A}\rangle p$ and its inverse;

(A-SNF3) $(\langle\overline{A}\rangle\langle\overline{A}\rangle\top \wedge \langle A\rangle\langle\overline{A}\rangle p) \rightarrow p \vee \langle\overline{A}\rangle\langle A\rangle\langle A\rangle p \vee \langle\overline{A}\rangle\langle\overline{A}\rangle\langle A\rangle p$ and its inverse;

(A-SNF4) $\langle A\rangle\langle A\rangle\langle A\rangle p \rightarrow \langle A\rangle\langle A\rangle p$ and its inverse.

Proposition 7. *A neighborhood frame* $\mathbf{F} = \langle \mathbb{I}, R, L\rangle$ *is an interval neighborhood frame if and only if the axioms A-SNF1, ..., A-SNF4 are valid in* $\mathbf{F}$.

Proof. As in Proposition 6. □

Note that the axioms cannot guarantee strictness of the neighborhood frame as irreflexivity is not definable in the language of PNL$^-$.

Theorem 25 (Soundness and Completeness). PNL$^-$ *is (sound and) complete for the class of all strict interval neighborhood structures.*

Proof. We closely follow the technique applied in the proof of Theorem 24. Again, the soundness is straightforward. For the completeness, we take any PNL$^-$-consistent formula ϕ. It is satisfied at the root w of some generated sub-model of the canonical model for PNL$^-$. We then pick a countable elementary sub-model $\mathbf{M}^{-*} = \langle \mathbf{F}^{-*}, V^*\rangle$ which contains w and satisfies ϕ there. Let $\mathbf{F}^{-*} = \langle \mathbb{I}^*, R^*, L^*\rangle$. Note that $\mathbf{F}^*$ is a weakly connected interval neighborhood frame in which the axioms A-SNF1, ..., A-SNF4 are valid since they are canonical (being of Sahlqvist type, up to tautological equivalence) and first-order definable. We then build step-by-step a model over a strict interval neighborhood structure, which maps p-morphically over $\mathbf{M}^{-*}$ very much like in the proof of Theorem 24, but easier, because we need not worry about point-intervals. □

8.2 Axiomatic Systems for Extensions of PNL^-

Theorem 26 (Soundness and Completeness). *We have the following completeness results:*

1. *the axiomatic system for* PNL^- *extended with A-SNF*^{u} *is sound and complete for* PNL^{u-}*;*
2. *the axiomatic system for* PNL^- *extended with A-SNF*^{de}*, is sound and complete for* PNL^{de-}*;*
3. *the axiomatic system for* PNL^- *extended with A-SNF*^{di} *is sound and complete for* PNL^{di-}*;*
4. *the axiomatic system for* PNL^- *combining* PNL^{u-} *and* PNL^{de-} *is sound and complete for* PNL^{ude-}*;*
5. *the axiomatic system for* PNL^- *combining* PNL^{u-} *and* PNL^{di-} *is sound and complete for* PNL^{udi-}*.*

Proof. All proofs are adaptations of the one for PNL^-, because the respective axioms are canonical and define semantic conditions which either are reflected by p-morphisms (unboundedness) or can be forced during the step-by step construction to hold in the limit structure (density or discreteness).

9 Semantic Tableau for $\mathcal{PNL}$

In this section we devise a classical tableau method for $\mathcal{PNL}$. We will do the work in detail for the logic PNL^+, and then we will present the necessary modifications for PNL^-. The method can also be adapted to include π. (In [48] we generalize the method to a large class of propositional interval temporal logics.)

First, some basic terminology. A **finite tree** is a finite directed connected graph in which every node, apart from one (the **root**), has exactly one incoming arc. A **successor** of a node $\mathbf{n}$ is a node $\mathbf{n'}$ such that there is an edge from $\mathbf{n}$ to $\mathbf{n'}$. A **leaf** is a node with no successors; a **path** is a sequence of nodes $\mathbf{n_1}, \ldots, \mathbf{n_k}$ such that, for all $j = 0, \ldots, k-1$, $\mathbf{n_{j+1}}$ is a successor of $\mathbf{n_j}$; a **branch** is a path from the root to a leaf. The **height** of a node $\mathbf{n}$ is the maximum length (number of edge) of a path from $\mathbf{n}$ to a leaf. If $\mathbf{n}, \mathbf{n'}$ belong to the same branch and the height of $\mathbf{n}$ is less than or equal to the height of $\mathbf{n'}$, we write $\mathbf{n} \prec \mathbf{n'}$. Let $\langle \mathbb{C}, < \rangle$ be a finite linear order. A **labeled formula**, with label in $\mathbb{C}$, is a pair $(\phi, [c_i, c_j])$, where ϕ is a formula in the language of PNL^+ ($\phi \in \mathrm{PNL}^+$ for short) and $[c_i, c_j] \in \mathbb{I}(\mathbb{C})^+$. For a node $\mathbf{n}$ in a tree, the **decoration** $\nu(\mathbf{n})$ is a triple $((\phi, [c_i, c_j]), \mathbb{C}, u_{\mathbf{n}})$, where $\langle \mathbb{C}, < \rangle$ is a finite linear order, $(\phi, [c_i, c_j])$ is a labeled formula, with label in $\mathbb{C}$, and $u_{\mathbf{n}}$ is a **local flag function** which associates the values 0 or 1 with every branch B containing $\mathbf{n}$. Intuitively, the value 0 for a node n with respect to a branch B means that n can be expanded on B. For the sake of simplicity, we will often assume the interval $[c_i, c_j]$ to consist of the elements $c_i < c_{i+1} < \cdots < c_j$, and sometimes, with a little abuse of notation, we will write $\mathbb{C} = \{c_1 < c_2 < \ldots\}$. A **decorated tree** is a tree in which every node has a decoration $\nu(\mathbf{n})$. For every decorated tree, we define a **global flag function** u acting on pairs *(node, branch through that node)* as $u(\mathbf{n}, B) = u_{\mathbf{n}}(B)$. Sometimes, for convenience, we will include in the decoration of the nodes the global flag function instead of the local ones. For any branch B in a decorated tree, we denote by $\mathbb{C}_B$ the ordered set in the decoration of the leaf B, and for any node $\mathbf{n}$ in a decorated tree, we denote by $\Phi(\mathbf{n})$ the formula in its decoration. If B is a branch, then $B \cdot \mathbf{n}$ denotes the result of the expansion of B with the node $\mathbf{n}$ (addition of an edge connecting the leaf of B to $\mathbf{n}$). Similarly, $B \cdot \mathbf{n_1} \mid \ldots \mid \mathbf{n_k}$ denotes the result of the expansion of B with k immediate successor nodes $\mathbf{n_1}, \ldots, \mathbf{n_k}$ (which produces k branches extending B). A tableau for PNL$^+$ will be defined as a special decorated tree. We note again that $\mathbb{C}$ remains finite throughout the construction of the tableau.

Definition 10. *Given a decorated tree $\mathcal{T}$, a branch B in $\mathcal{T}$, and a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\phi, [c_i, c_j]), \mathbb{C}, u)$, with $u(\mathbf{n}, B) = 0$, the* ***branch-expansion rule*** *for B and $\mathbf{n}$ is defined as follows (in all the considered cases, $u(\mathbf{n'}, B') = 0$ for all new pairs $(\mathbf{n'}, B')$ of nodes and branches).*

- *If $\phi = \neg\neg\psi$, then expand the branch to $B \cdot \mathbf{n_1}$, with $\nu(\mathbf{n_1}) = ((\psi, [c_i, c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \psi_0 \wedge \psi_1$, then expand the branch to $B \cdot \mathbf{n_1} \cdot \mathbf{n_2}$, with $\nu(\mathbf{n_1}) = ((\psi_0, [c_i, c_j]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_2}) = ((\psi_1, [c_i, c_j]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg(\psi_0 \wedge \psi_1)$, then expand the branch to $B \cdot \mathbf{n_1} | \mathbf{n_2}$, with $\nu(\mathbf{n_1}) = ((\neg\psi_0, [c_i, c_j]), \mathbb{C}_B, u)$ and $\nu(\mathbf{n_2}) = ((\neg\psi_1, [c_i, c_j]), \mathbb{C}_B, u_1)$.*
- *If $\phi = \Box_r\psi$ and c is the least element of $\mathbb{C}_B$, with $c_j \leq c$, which has not been used yet to expand $\mathbf{n}$ on B, then expand the branch to $B \cdot \mathbf{n_1}$ with $\nu(\mathbf{n_1}) = ((\psi, [c_j, c]), \mathbb{C}_B, u)$.*
- *If $\phi = \Box_l\psi$ and c is the greatest element of $\mathbb{C}_B$, with $c \leq c_i$, which has not been used yet to expand $\mathbf{n}$ on B, then expand the branch to $B \cdot \mathbf{n_1}$ with $\nu(\mathbf{n_1}) = ((\psi, [c, c_i]), \mathbb{C}_B, u)$.*
- *If $\phi = \neg\Box_r\psi$, then expand the branch to $B \cdot \mathbf{n_j} | \ldots | \mathbf{n_n} | \mathbf{n'_j} | \ldots | \mathbf{n'_n}$, where*
  1. *for all $j \leq k \leq n$, $\nu(\mathbf{n_k}) = ((\neg\psi, [c_j, c_k]), \mathbb{C}_B, u)$, and*
  2. *for all $j \leq k \leq n$, $\nu(\mathbf{n'_k}) = ((\neg\psi, [c_j, c]), \mathbb{C}_k, u)$, where, for $j \leq k \leq n-1$, $\mathbb{C}_k$ is the linear ordering obtained by inserting a new element c between c_k and c_{k+1} in $\mathbb{C}_B$, and, for $k = n$, $\mathbb{C}_k$ is the linear ordering obtained by inserting a new element c after c_n in $\mathbb{C}_B$.*
- *if $\phi = \neg\Box_l\psi$, then expand the branch to $B \cdot \mathbf{n_1} | \ldots | \mathbf{n_i} | \mathbf{n'_1} | \ldots | \mathbf{n'_i}$, where:*
  1. *for all $1 \leq k \leq i$, $\nu(\mathbf{n_k}) = ((\neg\psi, [c_k, c_i]), \mathbb{C}_B, u)$, and*
  2. *for all $1 \leq k \leq i$, $\nu(\mathbf{n'_k}) = ((\neg\psi, [c, c_i]), \mathbb{C}_k, u)$, where, for $2 \leq k \leq i$, $\mathbb{C}_k$ is the linear ordering obtained by inserting a new element c between c_{k-1} and c_k in $\mathbb{C}_B$, and, for $k = 1$, $\mathbb{C}_1$ is the linear ordering obtained by inserting a new element c before c_1 in $\mathbb{C}_B$.*

Finally, for any node $\mathbf{m}$ ($\neq \mathbf{n}$) in B and any branch B' extending B, let $u(\mathbf{m}, B')$ be equal to $u(\mathbf{m}, B)$, and for any branch B' extending B, $u(\mathbf{n}, B') = 1$, unless $\phi = \Box_l\psi$ or $\phi = \Box_r\psi$ (in such cases $u(\mathbf{n}, B') = 0$).

The universal formula $\Box_r\psi$ (the same holds for $\Box_l\psi$) states that, for all $c_j \leq c$, ψ holds over $[c_j, c]$. As a matter of fact, the expansion rule imposes such a condition for a single element c in $\mathbb{C}_B$ (the least element which has not been used yet), and it does not change the flag (which remains equal to 0). In this way, all elements will be eventually taken into consideration, including those elements greater than c_j that will be added to $\mathbb{C}_B$ in some subsequent steps of the tableau construction.
Let us define now the notions of open and closed branch. We say that a node $\mathbf{n}$ in a decorated tree $\mathcal{T}$ is **available on a branch** B to which it belongs if and only if $u(\mathbf{n}, B) = 0$. The branch-expansion rule is **applicable** to a node $\mathbf{n}$ on a branch B if the node is available on B and the application of the rule generates at least one successor node with a new labeled formula. This second condition is needed to avoid looping of the application of the rule on universal formulas.

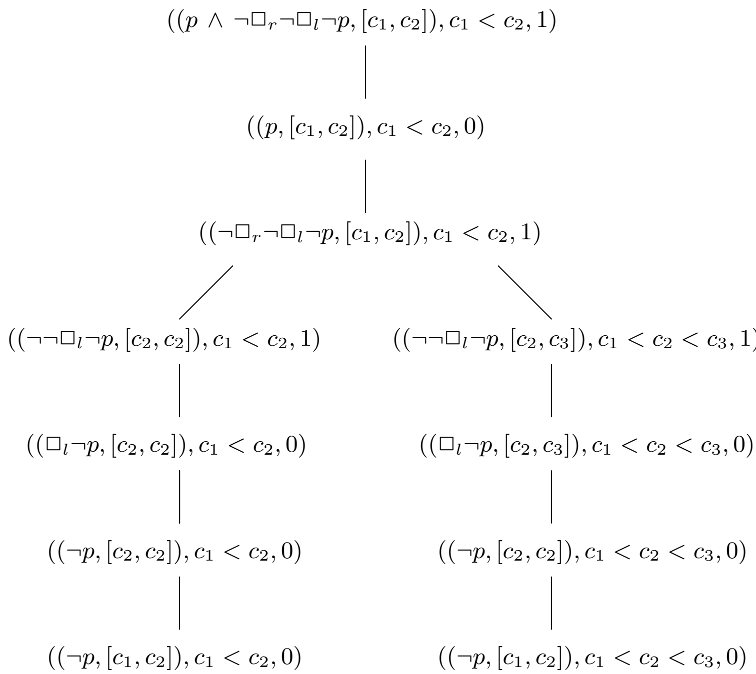


Fig. 2. Example of the tableau method for PNL$^+$

Definition 11. *A branch B in a tableau for ϕ is* ***closed*** *if and only if there are two nodes $\mathbf{n}, \mathbf{n}' \in B$ such that $\nu(\mathbf{n}) = ((\psi, [c_i, c_j]), \mathbb{C}, u)$ and $\nu(\mathbf{n}') = ((\neg\psi, [c_i, c_j]), \mathbb{C}', u)$ for some formula ψ and $c_i, c_j \in \mathbb{C} \cap \mathbb{C}'$, otherwise it is* ***open****.*

Definition 12. *The* ***branch-expansion strategy*** *for a branch B in a decorated tree $\mathcal{T}$ is as follows: (1) apply the branch-expansion rule to a branch B only if it is open; (2) if B is open, apply the branch-expansion rule to B on the closest to the root available node for which the branch-expansion rule is applicable.*

Definition 13. *A* ***tableau*** *for a given formula $\phi \in$PNL$^+$ is any finite decorated tree $\mathcal{T}$ obtained by expanding the one-node decorated tree $((\phi, [c_1, c_2]), \{c_1, c_2\}, u)$, where the (only) value of u is 0, through successive applications of the branch-expansion strategy to currently existing branches.*

Definition 14. *A tableau for a given formula $\phi \in$PNL$^+$ is* ***closed*** *if and only if every branch in it is closed, otherwise it is* ***open****.*

In Figure 2 we showed the case of the tableau for the formula $p \wedge \neg\Box_r\neg\Box_l\neg p$, that is, $\neg$ A-NNF1. As one can expect, all branches of the tableau are closed, meaning that the formula is not satisfiable.

9.1 Soundness and Completeness

Definition 15. *Given a set S of labeled formulas with labels in a linear ordering $\langle \mathbb{C}, < \rangle$, we say that S is* ***satisfiable over*** *$\mathbb{C}$ if there exists a non-strict model $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{D})^+, R, L, V \rangle$ such that $\langle \mathbb{D}, < \rangle$ is an extension of $\langle \mathbb{C}, < \rangle$ and $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$ for all $(\psi, [c_i, c_j]) \in S$.*

Clearly, the above notion is equivalent to the notion of satisfiability of a formula in the case that S contains only one labeled formula.

Theorem 27 (Soundness). *If $\phi \in$PNL$^+$ and a tableau $\mathbf{T}$ for ϕ is closed, then ϕ is not satisfiable.*

Proof. We prove by induction on the height h of a node $\mathbf{n}$ in the tableau $\mathbf{T}$ a stronger claim: if every branch containing $\mathbf{n}$ is closed, then the set $S(\mathbf{n})$ of all labeled formulas in the decorations of the nodes between $\mathbf{n}$ and the root is not satisfiable over $\mathbb{C}$, where $\mathbb{C}$ is the linear ordering in the decoration of $\mathbf{n}$.

If $h = 0$, then $\mathbf{n}$ is a leaf and the unique branch B containing $\mathbf{n}$ is closed. Then $S(\mathbf{n})$ contains $(\psi, [c_k, c_l])$ and $(\neg\psi, [c_k, c_l])$ for some PNL$^+$-formula ψ. Take any model $\mathbf{M}^+ = \langle(\mathbb{I}(\mathbb{D})^+, R, L, V\rangle$, where $\langle\mathbb{D}, <\rangle$ is an extension of $\langle\mathbb{C}, <\rangle$. $\mathbf{M}^+, [c_k, c_l] \vDash \psi$ iff $\mathbf{M}^+, [c_k, c_l] \nVdash \neg\psi$, and thus $S(\mathbf{n})$ is not satisfiable over $\mathbb{C}$.

Suppose $h > 0$. Then either the branch-expansion rule has been applied to some labeled formula $(\psi, [c_i, c_j]) \in S(\mathbf{n})$ to extend the branch at $\mathbf{n}$, or $\mathbf{n}$ has been generated as the first of the two successors obtained by an application of a $\wedge$-rule. We will only consider in detail the former case, as the latter is subsumed by it.

Let $\mathbb{C} = \{c_1, \ldots, c_n\}$, where $c_1 < \ldots < c_n$, be the linear ordering from the decoration of $\mathbf{n}$. Note that every branch passing through any successor of $\mathbf{n}$ must be closed, so the inductive hypothesis applies to all successors of $\mathbf{n}$.

We consider the possible cases for the branch-expansion rule applied at $\mathbf{n}$.

- Let $\psi = \neg\neg\xi$. Then there exists $\mathbf{n_1}$ such that $\nu(\mathbf{n_1}) = ((\xi, [c_i, c_j]), \mathbb{C}, u)$ and $\mathbf{n_1}$ is a successor of $\mathbf{n}$. Since every branch containing $\mathbf{n}$ is closed, then every branch containing $\mathbf{n_1}$ is closed. By the inductive hypothesis, $S(\mathbf{n_1})$ is not satisfiable over C (since $\mathbf{n_1} \prec \mathbf{n}$). Since ξ_0 and $\neg\neg\xi_0$ are equivalent, $S(\mathbf{n})$ cannot be satisfiable over $\mathbb{C}$.
- Let $\psi = \xi_1 \wedge \xi_2$. Then there are two nodes $\mathbf{n_1} \in B$ and $\mathbf{n_2} \in \mathbf{B}$ such that $\nu(\mathbf{n_1})$ $= ((\xi_1, [c_i, c_j]), \mathbb{C}, u)$, $\nu(\mathbf{n_2}) = ((\xi_2, [c_i, c_j]), \mathbb{C}, u)$, and, without loss of generality, $\mathbf{n_1}$ is the successor of $\mathbf{n}$ and $\mathbf{n_2}$ is the successor of $\mathbf{n_1}$. Since every branch containing $\mathbf{n}$ is closed, then every branch containing $\mathbf{n_2}$ is closed. By the inductive hypothesis, $S(\mathbf{n_2})$ is not satisfiable over C since $\mathbf{n_2} \prec \mathbf{n}$. Since every model over $\mathbb{C}$ satisfying $S(\mathbf{n})$ must, in particular, satisfy $(\xi_1 \wedge \xi_2, [c_i, c_j])$, and hence $(\xi_1, [c_i, c_j])$ and $(\xi_2, [c_i, c_j])$, it follows that $S(\mathbf{n})$, $S(\mathbf{n_1})$, and $S(\mathbf{n_2})$ are equi-satisfiable over $\mathbb{C}$. Therefore, $S(\mathbf{n})$ is not satisfiable over $\mathbb{C}$.
- let $\psi = \neg(\xi_1 \wedge \xi_2)$. Then there exist two successor nodes $\mathbf{n_1}$ and $\mathbf{n_2}$ of $\mathbf{n}$, such that $\nu(\mathbf{n_1}) = ((\xi_0, [c_i, c_j]), \mathbb{C}, u_0)$, $\nu(\mathbf{n_2}) = ((\xi_1, [c_i, c_j]), \mathbb{C}, u_1)$, $\mathbf{n_1}, \mathbf{n_2} \prec \mathbf{n}$. Since every branch containing $\mathbf{n}$ is closed, then every branch containing $\mathbf{n_1}$ and every branch containing $\mathbf{n_2}$ is closed. By the inductive hypothesis, $S(\mathbf{n_1})$ and $S(\mathbf{n_2})$ are not satisfiable over $\mathbb{C}$. Since every model over $\mathbb{C}$ satisfying $S(\mathbf{n})$ must also satisfy $(\xi_0, [c_i, c_j])$ or $(\xi_1, [c_i, c_j])$, it follows that $S(\mathbf{n})$ cannot be satisfiable over $\mathbb{C}$.
- Let $\psi = \neg\Box_r\xi$. Assuming that $S(\mathbf{n})$ is satisfiable over $\mathbb{C}$, there is a model $\mathbf{M}^+ = \langle(\mathbb{I}(\mathbb{D})^+, R, L, V\rangle$, where $\langle\mathbb{D}, <\rangle$ is an extension of $\langle\mathbb{C}, <\rangle$, such that $\mathbf{M}^+, [c_i, c_j] \Vdash \theta$ for all $(\theta, [c_i, c_j]) \in S(\mathbf{n})$. In particular, $\mathbf{M}^+, [c_j, d] \vDash \neg\xi$ for some $d \geq c_j$. Consider 2 cases:
  1. $d \in \mathbb{C}$. Then $d = c_m$ for some $m \geq j$. But one of the successor nodes of $\mathbf{n}$ is $\mathbf{n_m}$, where $\nu(\mathbf{n_m}) = ((\neg\xi, [c_j, c_m]), \mathbb{C}, u)$, and since $\mathbf{n_m} \prec \mathbf{n}$,, by the inductive hypothesis, $S(\mathbf{n_m}) = S(\mathbf{n}) \cup\{(\neg\xi, [c_j, c_m])\}$ is not satisfiable over $\mathbb{C}$, which is a contradiction.

2. $d \notin \mathbb{C}$. Then there is an m such that $j \leq m \leq n-1$ and $c_m < d < c_{m+1}$, or $m = n$ and $c_n < d$. In either case, there is a successor node $\mathbf{n'_m}$ of $\mathbf{n}$ such that $\nu(\mathbf{n'_m})$ $= ((\neg\xi, [c_j, d]), \mathbb{C} \cup \{d\}, u)$, and since $\mathbf{n'_m} \prec \mathbf{n}$, by the inductive hypothesis $S(\mathbf{n'_m})$ $= S(\mathbf{n}) \cup \{(\neg\xi, [c_j, d])\}$ is not satisfiable over $\mathbb{C} \cup \{d\}$, which, again, is a contradiction.

Thus, in either case $S(\mathbf{n})$ is not satisfiable over $\mathbb{C}$.

- The case of $\psi = \neg\Box_l\xi$ is analogous.
- Let $\psi = \Box_r\xi$. Then $\nu(\mathbf{n_1}) = ((\xi, [c_j, c_m]), \mathbb{C}, u)$, with $j \leq m \leq n$, for the successor $\mathbf{n_1}$ of $\mathbf{n}$. Now, any model over $\mathbb{C}$ satisfying $S(\mathbf{n_1})$ must, in particular, satisfy $(\Box_r\xi, [c_i, c_j])$, and hence $(\xi, [c_j, c_m])$. Thus, the sets $S(\mathbf{n})$ and $S(\mathbf{n_1}) = \mathbf{S(n)} \cup \{(\xi, [\mathbf{c_j}, \mathbf{c_m}])\}$ are equi-satisfiable over $\mathbb{C}$. Since $\mathbf{n_1} \prec \mathbf{n}$, by the inductive hypothesis $S(\mathbf{n_1})$ is not satisfiable over $\mathbb{C}$, and thus $S(\mathbf{n})$ is not satisfiable over $\mathbb{C}$.
- The case of $\psi = \Box_l\xi$ is analogous. □

Definition 16. *If $\mathbf{T}_0$ is the one-node tableau $((\phi, [c_1, c_2]), \{c_1, c_2\}, 0)$ for a given* PNL$^+$-*formula ϕ, the **limit tableau** $\overline{\mathbf{T}}$ for ϕ is the (possibly infinite) decorated tree obtained as follows. First, for all i, $\mathbf{T}_{i+1}$ is the tableau obtained by simultaneous application of the branch-expansion strategy to every branch in $\mathbf{T}_i$. Then, we ignore all flags from the decorations of the nodes in every $\mathbf{T}_i$. Thus we obtain a chain by inclusion of decorated trees: $\mathbf{T}_0 \subseteq \mathbf{T}_1 \subseteq \ldots$. Now we define $\overline{\mathbf{T}} := \bigcup_{i=0}^{\infty} \mathbf{T}_i$.*

Note that the chain above may stabilize at some $\mathbf{T}_i$ if it closes, or if the branch-expansion rule is not applicable to any of its branches. We associate with each branch B in $\overline{\mathbf{T}}$ the linear ordering $\mathbb{C}_B = \bigcup_{i=0}^{\infty} \mathbb{C}_{B_i}$, where, for all i, $\mathbb{C}_{B_i}$ is the linear ordering from the decoration of the leaf of the (sub-)branch B_i of B in $\mathbf{T}_i$. The definitions of closed and open branches readily apply to $\overline{\mathbf{T}}$.

Definition 17. *A branch in a (limit) tableau is **saturated** if there are no nodes on it to which the branch-expansion rule is applicable on the branch. A (limit) tableau is **saturated** if every open branch in it is saturated.*

In what follows we will show that the set of all labeled formulas on an open branch in a limit tableau has the saturation properties of a Hintikka set in first-order logic.

Lemma 3. *Every limit tableau is saturated.*

Proof. Given a node $\mathbf{n}$ in a limit tableau $\overline{\mathbf{T}}$, we denote by $d(\mathbf{n})$ the distance (number of edges) between $\mathbf{n}$ and the root of $\overline{\mathbf{T}}$. Now, given a branch B in $\overline{\mathbf{T}}$, we will prove by induction on $d(\mathbf{n})$ that after every step of the expansion of that branch at which the branch-expansion rule becomes applicable to $\mathbf{n}$ (because $\mathbf{n}$ has just been introduced, or because a new point has been introduced in the linear ordering on B) that rule is subsequently applied on B to that node.
Suppose the inductive hypothesis holds for all nodes with distance to the root less than m. Let $d(\mathbf{n}) = m$ and the branch-expansion rule has become applicable to $\mathbf{n}$. If there are no nodes between the root (including the root) and $\mathbf{n}$ (excluding $\mathbf{n}$) to which the branch-expansion rule is applicable at that moment, the next application of the branch-expansion rule on B is to $\mathbf{n}$. Otherwise, consider the closest to $\mathbf{n}$ node $\mathbf{n}^*$ between the root and $\mathbf{n}$ to which the branch-expansion rule is applicable, or becomes applicable on B at least once thereafter. (Such node exists because there are only finitely many nodes between $\mathbf{n}$ and the

root.) Since $d(\mathbf{n}^*) < d(\mathbf{n})$, by the inductive hypothesis the branch-expansion rule has been subsequently applied to $\mathbf{n}^*$. Then the next application of the branch-expansion rule on B must have been to $\mathbf{n}$ and that completes the induction. Now, assuming that a branch in a limit tableau is not saturated, consider the closest to the root node $\mathbf{n}$ on that branch B to which the branch-expansion rule is applicable on that branch. If $\Phi(\mathbf{n})$ is neither $\Box_r\psi$ nor $\Box_l\psi$, then the branch-expansion rule has become applicable to $\mathbf{n}$ at the step when $\mathbf{n}$ is introduced, and by the claim above, it has been subsequently applied, at which moment the node has become unavailable thereafter, which contradicts the assumption. If $\Phi(\mathbf{n}) =\Box_r\psi$ or $\Phi(\mathbf{n}) =\Box_l\psi$, then an application of the rule on B must create at least one successor with a new label $(\psi, [c_i, c_j])$ on B. But c_i, c_j have already been introduced at some (finite) step of the construction of B and at the first step when both of them, as well as $\mathbf{n}$, have appeared on the branch, the branch-expansion rule has become applicable to $\mathbf{n}$, hence is has been subsequently applied on B and that application must have introduced the label $(\psi, [c_i, c_j])$ on B, which again contradicts the assumption. □

Corollary 6. *Let ϕ be a* PNL$^+$*-formula, and $\overline{\mathbf{T}}$ the limit tableau for ϕ. Then, for every open branch B in $\overline{\mathbf{T}}$:*

- *if there is a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\neg\neg\psi, [c_i, c_j]), \mathbb{C}, u)$, then there is a node $\mathbf{n_1} \in B$ such that $\nu(\mathbf{n_1}) = ((\psi, [c_i, c_j]), \mathbb{C}, u_1)$;*
- *if there is a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\psi_1 \wedge \psi_2, [c_i, c_j]), \mathbb{C}, u)$, then there is a node $\mathbf{n_1} \in B$ such that $\nu(\mathbf{n_1}) = ((\psi_1, [c_i, c_j]), \mathbb{C}, u_1)$ and a node $\mathbf{n_2} \in B$ such that $\nu(\mathbf{n_2}) = ((\psi_2, [c_i, c_j]), \mathbb{C}, u_2)$;*
- *if there is a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\neg(\psi_1 \wedge \psi_2), [c_i, c_j]), \mathbb{C}, u)$, then there is a node $\mathbf{n_1} \in B$ such that $\nu(\mathbf{n_1}) = ((\neg\psi_1, [c_i, c_j]), \mathbb{C}, u_1)$ or a node $\mathbf{n_2} \in B$ such that $\nu(\mathbf{n_2}) = ((\neg\psi_2, [c_i, c_j]), \mathbb{C}, u_2)$;*
- *if there is a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\neg\Box_r\psi, [c_i, c_j]), \mathbb{C}, u)$, then, for some $c \in \mathbb{C}_B$ such that $c_j \leq c$ there is a node $\mathbf{n}' \in B$ such that $\nu(\mathbf{n}') = ((\neg\psi, [c_j, c]), \mathbb{C}', u')$;*
- *likewise for every node $\mathbf{n}$ with $\Phi(\mathbf{n}) = \neg\Box_l\psi$;*
- *if there is a node $\mathbf{n} \in B$ such that $\nu(\mathbf{n}) = ((\Box_r\psi, [c_i, c_j]), \mathbb{C}, u)$, then for all $c \in \mathbb{C}_B$ such that $c_j \leq c$, there is a node $\mathbf{n}' \in B$ such that $\nu(\mathbf{n}') = ((\psi, [c_j, c]), \mathbb{C}', u')$;*
- *likewise for every node $\mathbf{n}$ with $\Phi(\mathbf{n}) = \Box_l\psi$.*

Lemma 4. *If the limit tableau for some formula $\phi \in$*PNL$^+$ *is closed, then some finite tableau for ϕ is closed.*

Proof. Suppose the limit tableau for ϕ is closed. Then every branch closes at some finite step of the construction and then remains finite. Since the branch-expansion rule always produces finitely many successors, every finite tableau is finitely branching, and hence so is the limit tableau. Then, by König's lemma, the limit tableau, being a finitely branching tree with no infinite branches, must be finite, hence its construction stabilizes at some finite stage. At that stage a closed tableau for ϕ is constructed. □

Theorem 28 (Completeness). *Let $\phi \in$*PNL$^+$ *be a valid formula. Then there is a closed tableau for $\neg\phi$.*

Proof. We will show that the limit tableau $\overline{\mathbf{T}}$ for $\neg\phi$ is closed, whence the theorem follows by the previous lemma.

By contraposition, suppose that $\overline{\mathbf{T}}$ has an open branch B. Let $\mathbb{C}_B$ be the linear ordering associated with B and $\Phi(B)$ be the set of all labeled formulas on B. Consider the model $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{C}_B)^+, R, L, V \rangle$ where, for every $[c_i, c_j] \in \mathbb{I}(\mathbb{C}_B)^+$ and $p \in \mathcal{AP}$,

$$p \in V([c_i, c_j]) \text{ iff } (p, [c_i, c_j]) \in \Phi(B).$$

We are going to show by induction on ψ that, for every $(\psi, [c_i, c_j]) \in \Phi(B)$,

$$\mathbf{M}^+, [c_i, c_j] \Vdash \psi.$$

1. If $\psi = p$ or $\psi = \neg p$ where $p \in \mathcal{AP}$, the claim follows by definition, because if $(\neg p, [c_i, c_j]) \in \Phi(B)$, then $(p, [c_i, c_j]) \notin \Phi(B)$ since B is open (the same for $(p, [c_i, c_j]) \in \Phi(B)$).
2. Let $\psi = \neg\neg\xi$. Then by Lemma 6, $(\xi, [c_i, c_j]) \in \Phi(B)$, and by inductive hypothesis $\mathbf{M}^+, [c_i, c_j] \Vdash \xi$. So $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$.
3. Let $\psi = \xi_0 \wedge \xi_1$. Then by Lemma 6, $(\xi_0, [c_i, c_j]) \in \Phi(B)$ and $(\xi_1, [c_i, c_j]) \in \Phi(B)$. By inductive hypothesis, $\mathbf{M}^+, [c_i, c_j] \Vdash \xi_0$ and $\mathbf{M}^+, [c_i, c_j] \Vdash \xi_1$, so $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$.
4. Let $\psi = \neg(\xi_0 \wedge \xi_1)$. Then by Lemma 6, $(\neg\xi_0, [c_i, c_j]) \in \Phi(B)$ or $(\neg\xi_1, [c_i, c_j]) \in \Phi(B)$. By inductive hypothesis $\mathbf{M}^+, [c_i, c_j] \Vdash \neg\xi_0$ or $\mathbf{M}^+, [c_i, c_j] \Vdash \neg\xi_1$, so $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$.
5. Let $\psi = \neg\Box_r\xi$. Then by Lemma 6, $(\neg\xi, [c_j, c]) \in \Phi(B)$ for some $c \in \mathbb{C}_B$ such that $c_j \leq c$. Thus, by inductive hypothesis, $\mathbf{M}^+, [c_j, c] \Vdash \neg\xi$. So, $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$.
6. The case of $\psi = \neg\Box_l\xi$ is similar.
7. Let $\psi = \Box_r\xi$. Then by Lemma 6, $(\xi, [c_j, c]) \in \Phi(B)$ for all $c \in \mathbb{C}_B$ such that $c_j \leq c$. Hence, by inductive hypothesis, for all $c \in \mathbb{C}_B$ such that $c_j \leq c$ $\mathbf{M}^+, [c_j, c] \Vdash \xi$, and so $\mathbf{M}^+, [c_i, c_j] \Vdash \psi$.
8. The case of $\psi = \Box_l\xi$ is similar.

This completes the induction. In particular, we obtain that $\neg\phi$ is satisfied in $\mathbf{M}^+$, which is in contradiction with the assumption that ϕ is valid. □

The tableau method for PNL$^+$ developed here can be easily adapted to the case of PNL$^-$. Indeed, the method requires the following straightforward modification of the branch-expansion rule: in the cases of $\neg[A]$, $\neg[\overline{A}]$, $[A]$, and $[\overline{A}]$ the rule does not introduce a successor node with label $[c_j, c_j]$. With that modification, all theorems and their proofs included in this section can be accordingly adapted for PNL$^-$.

10 Conclusion and Possible Developments

In this paper we have studied the class of Propositional Neighborhood Logics ($\mathcal{PNL}$), and we have provided complete axiomatic systems and a classical tableau method for them. Currently the questions about the decidability of $\mathcal{PNL}$ logics are open (as a matter of fact, the technique suggested by Montanari and Sciavicco in [80], as it stands, does not work). Generally speaking, the problem of finding decidable fragments of interval temporal logics has been raised by several authors, including Halpern and Shoham (cf. Problem 4 in [59]) and Venema (cf. Question 3.20 in [110]). Possible approaches to prove decidability are via finite model property or interpretation into other decidable logics. Note, however, that the finite model property with respect to standard models fails in both semantics [49], and thus one could only hope for a finite model property with respect to non-standard models.

Copy of: **D. Bresolin, A. Montanari, and G. Sciavicco: An Optimal Decision Procedure for Right Propositional Neighborhood Logic. Journal of Automatic Reasoning, 38(1-3):173–199, 2007.**

Abstract. Propositional interval temporal logics are quite expressive temporal logics that allow one to naturally express statements that refer to time intervals. Unfortunately, most such logics turned out to be (highly) undecidable. To get decidability, severe syntactic and/or semantic restrictions have been imposed to interval-based temporal logics that make it possible to reduce them to point-based ones. The problem of identifying expressive enough, yet decidable, new interval logics or fragments of existing ones which are genuinely interval-based is still largely unexplored. In this paper, we focus our attention on interval logics of temporal neighborhood. We address the decision problem for the future fragment of Neighborhood Logic (Right Propositional Neighborhood Logic, RPNL for short) and we positively solve it by showing that the satisfiability problem for RPNL over natural numbers is NEXPTIME-complete. Then, we develop a sound and complete tableau-based decision procedure and we prove its optimality.

1 Introduction

Propositional interval temporal logics are quite expressive temporal logics that provide a natural framework for representing and reasoning about temporal properties in several areas of computer science, including artificial intelligence (reasoning about action and change, qualitative reasoning, planning, and natural language processing), theoretical computer science (specification and automatic verification of programs) and databases (temporal and spatio-temporal databases).

Various propositional and first-order interval temporal logics have been proposed in the literature (a recent comprehensive survey can be found in [52]). The most significant propositional ones are Halpern and Shoham's Modal Logic of Time Intervals (HS) [59], Venema's CDT logic, interpreted over linear and partial orders [48,53,110], Moszkowski's Propositional Interval Temporal Logic (PITL) [84], and Goranko, Montanari, and Sciavicco's Propositional Neighborhood Logic (PNL) [50].

HS features four basic operators: $\langle B\rangle$ (*begins*) and $\langle E\rangle$ (*ends*), and their transposes $\langle\overline{B}\rangle$ and $\langle\overline{E}\rangle$. Given a formula φ and an interval $[d_0, d_1]$, $\langle B\rangle\varphi$ holds at $[d_0, d_1]$ if φ holds at $[d_0, d_2]$, for some $d_2 < d_1$, and $\langle E\rangle\varphi$ holds at $[d_0, d_1]$ if φ holds at $[d_2, d_1]$, for some $d_2 > d_0$. HS has been shown to be undecidable for several classes of linear and branching orders, including natural numbers [59]. The fragment of HS with the two modalities $\langle B\rangle$ and $\langle E\rangle$ only has been proved to be undecidable when interpreted over dense linear orders by sharpening the original Halpern and Shoham's result [74].

CDT has three binary operators C (*chop*), D, and T, which correspond to the ternary interval relations occurring when an extra point is added in one of the three possible distinct positions with respect to the two endpoints of the current interval (*before, between*, and *after*), plus a modal constant π which holds at a given interval if and only if it is a point-interval. CDT is powerful enough to embed HS, and thus it is undecidable (at least) over the same classes of orders.

PITL provides two modalities, namely, $\bigcirc$ (*next*) and C (the specialization of the *chop* operator for discrete structures). In PITL an interval is defined as a finite or infinite sequence of states. Given two formulas φ, ψ and an interval $s_0, \ldots, s_n$, $\bigcirc\varphi$ holds over $s_0, \ldots, s_n$ if and only if φ holds over $s_1, \ldots, s_n$, while $\varphi\, C\, \psi$ holds over $s_0, \ldots, s_n$ if and only if there exists i, with $0 \leq i \leq n$, such that φ holds over $s_0, \ldots, s_i$ and ψ holds over $s_i, \ldots, s_n$. In [84], Moszkowski proves the undecidability of PITL over discrete linear orders, while its undecidability over dense linear orders has been shown by Lodaya [74].

PNL has two modalities for right and left interval neighborhoods, namely, the *after* operator $\langle A \rangle$, such that $\langle A \rangle \varphi$ holds over $[d_0, d_1]$ if φ holds over $[d_1, d_2]$ for some $d_2 > d_1$, and its transpose $\langle \overline{A} \rangle$ [50]. While the undecidability of first-order Neighborhood Logic (NL) can be easily proved by embedding HS in it, the satisfiability problem for PNL has been recently shown to be decidable in NEXPTIME with respect to several class of linear orders [81]. The proof basically reduces the problem to the satisfiability problem for the decidable 2-variable fragment of first-order logic extended with a linear order [92]. As a matter of fact, such a reduction does not provide us with a viable decision procedure for PNL.

In summary, propositional interval temporal logics are very expressive (it can be shown that both HS and CDT are strictly more expressive than every point-based temporal logic on linear orders), but in general (highly) undecidable. They make it possible to express properties of *pairs* of time points (think of intervals as constructed out of points), rather than *single* time points, and, in most cases, this feature prevents one from the possibility of reducing interval-based temporal logics to (decidable) point-based ones and of benefitting from the good computational properties of point-based logics.

To make such a reduction possible, severe syntactic and/or semantic restrictions must be imposed to interval temporal logics [79].

One can get decidability by making a suitable choice of the interval modalities. This is the case with the $\langle B \rangle \langle \overline{B} \rangle$ (*begins/begun by*) and and $\langle E \rangle \langle \overline{E} \rangle$ (*ends/ended by*) fragments of HS. Consider the case of $\langle B \rangle \langle \overline{B} \rangle$ (the case of $\langle E \rangle \langle \overline{E} \rangle$ is similar). As shown by Goranko et al. [52], the decidability of $\langle B \rangle \langle \overline{B} \rangle$ can be obtained by embedding it into the propositional temporal logic of linear time LTL[F,P] with temporal modalities F (sometime in the future) and P (sometime in the past). The formulae of $\langle B \rangle \langle \overline{B} \rangle$ are simply translated into formulae of LTL[F,P] by a mapping that replaces $\langle B \rangle$ by P and $\langle \overline{B} \rangle$ by F. LTL[F,P] has the finite model property and is decidable.

As an alternative, decidability can be achieved by constraining the classes of temporal structures over which the interval logic is interpreted. This is the case with the so-called Split Logics (SLs) [82]. SLs are propositional interval logics equipped with operators borrowed from HS and CDT, but interpreted over specific structures, called split structures. The distinctive feature of split structures is that every interval can be 'chopped' in at most one way. The decidability of various SLs has been proved by embedding them into first-order fragments of monadic second-order decidable theories of time granularity (which are proper extensions of the well-known monadic second-order theory of one successor S1S).

Finally, another possibility is to constrain the relation between the truth value of a formula over an interval and its truth value over subintervals of that interval. As an example, one can constrain a propositional letter to be true over an interval if and only if it is true at its starting point (*locality*) or can constrain it to be true over an interval if and only it it is true over all its subintervals (*homogeneity*). A decidable fragment of PITL extended with quantification over propositional letters (QPITL) has been obtained by imposing the *locality* constraint [84]. By exploiting such a constraint, decidability of QPITL can be proved by embedding it into quantified LTL. (In fact, as already noticed by Venema, the locality assumption yields decidability even in the case of the interval logics HS and CDT [110].)

A major challenge in the area of propositional interval temporal logics is thus to identify *genuinely interval-based* decidable logics, that is, logics which are not explicitly translated into point-based logics and not invoking locality or other semantic restrictions, and to provide them with actual decision procedures.

In this paper, we propose an implicit and incremental tableau-based decision procedure for the future fragment of PNL, that we call Right PNL (RPNL for short), interpreted over natural numbers. While various tableau methods have been developed for linear and branching time point-based temporal logics [40,41,71,104,112], not much work has been done on tableau methods for interval-based temporal logics. One reason for this disparity is that operators of interval temporal logics are in many respects more difficult to deal with [53]. As an example, there exist straightforward inductive definitions of the basic operators of point-based temporal logics, while inductive definitions of interval modalities turn out to be much more involved (consider, for instance, the one for the *chop* operator given in [11]). In [48,53], Goranko et al. propose a general tableau method for CDT, interpreted over partial orders. It combines features of the classical tableau method for first-order logic with those of explicit tableau methods for modal logics with constraint label management, and it can be easily tailored to most propositional interval temporal logics proposed in the literature. However, it only provides a semi-decision procedure for unsatisfiability.
In this paper, we devise an optimal tableaux-based decision procedure for RPNL. Unlike the case of the $\langle B\rangle\langle\overline{B}\rangle$ and $\langle E\rangle\langle\overline{E}\rangle$ fragments, we cannot abstract way from the left endpoint of intervals: there can be contradictory formulae that hold over intervals that have the same right endpoint, but a different left one. The proposed tableau method partly resembles the tableau-based decision procedure for LTL [112]. However, while the latter takes advantage of the so-called fix-point definition of temporal operators, which makes it possible to proceed by splitting every temporal formula into a (possibly empty) part related to the current state and a part related to the next state, and to completely forget the past, our method must also keep track of universal and (pending) existential requests coming from the past.

The paper is organized as follows. In Section 2, we introduce syntax and semantics of RPNL. We distinguish two possible semantics, namely, a *strict* one, which excludes intervals with coincident endpoints (*point-intervals*), and a *non-strict* one, which includes them. In Section 3, we give an intuitive account of the proposed decision method, in the case of strict semantics, and then, in Section 4, we formalize it. In Section 5 we prove the NEXPTIME-completeness of the satisfiability problem for RPNL, while in Section 6 we devise an optimal tableau-based decision procedure and we prove its soundness and completeness. Finally, in Section 7 we briefly show how to adapt the method to the case of non-strict semantics. Conclusions provide an assessment of the work and outline future research directions.

2 Right Propositional Neighborhood Logics

In this section, we give syntax and semantics of RPNL interpreted over the set $\mathbb{N}$ of natural numbers or over a prefix of it. To this end, we introduce some preliminary notions. Let $\mathbb{D} = \langle D, <\rangle$ be a strict linear order isomorphic to $\mathbb{N}$ or to a prefix of it. An *interval* on $\mathbb{D}$ is an ordered pair $[d_i, d_j]$ such that $d_i, d_j \in D$ and $d_i \leq d_j$. We say that $[d_i, d_j]$ is a *strict interval* if $d_i < d_j$, and that it is a *point interval* if $d_i = d_j$. The set of all strict intervals will be denoted by $\mathbb{I}(\mathbb{D})^-$, while the set of all intervals on $\mathbb{D}$ will be denoted by $\mathbb{I}(\mathbb{D})^+$. With $\mathbb{I}(\mathbb{D})$ we denote either of these. The pair $\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle$ is called a *strict interval structure*, while the pair $\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^+\rangle$ is called a *non-strict interval structure*. For every pair of intervals $[d_i, d_j], [d'_i, d'_j] \in \mathbb{I}(\mathbb{D})$, we say that $[d'_i, d'_j]$ is a *right neighbor* of $[d_i, d_j]$ if and only if $d_j = d'_i$.

The language of *Strict Right Propositional Neighborhood Logic* (RPNL^- for short) consists of a set AP of propositional letters, the classical connectives $\neg$ and $\vee$, and the modal operator $\langle A\rangle$, the dual of which is denoted $[A]$. The remaining classical connectives, as well as the logical constants $\top$ (true) and $\bot$ (false), can be defined as usual. The *formulae* of RPNL^-, denoted by $\varphi, \psi, \ldots$, are recursively defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A\rangle\varphi.$$

The language of *Non-strict Right Propositional Neighborhood Logic* (RPNL^+ for short) differs from the language of RPNL^- only in the notation for the modalities: $\langle A\rangle$ and $[A]$ are replaced by $\Diamond_r$ and $\Box_r$, respectively. We use different notations for the modalities in RPNL^- and RPNL^+ only to reflect their historical links and to make it easier to distinguish between the two semantics from the syntax. The *formulae* of RPNL^+, are recursively defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_r\varphi.$$

We denote by $|\varphi|$ the length of φ, that is, the number of symbols in φ (in the following, we shall use $|\ |$ to denote the cardinality of a set as well). Whenever there are no ambiguities, we call an RPNL formula just a formula. A formula of the forms $\langle A\rangle\psi$ or $\neg\langle A\rangle\psi$ (resp., $\Diamond_r\psi$ or $\neg\Diamond_r\psi$) is called a *temporal formula* (from now on, we identify $\neg\langle A\rangle\psi$ with $[A]\neg\psi$ and $\neg\Diamond_r\psi$ with $\Box_r\neg\psi$).

A *model* for an RPNL^- (resp., RPNL^+) formula is a pair $\mathbf{M} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})\rangle, \mathcal{V}\rangle$, where $\langle\mathbb{D}, \mathbb{I}(\mathbb{D})\rangle$ is a strict (resp., non-strict) interval structure and $\mathcal{V} : \mathbb{I}(\mathbb{D}) \longrightarrow 2^{AP}$ is a *valuation function* assigning to every interval the set of propositional letters true on it. Given a model $\mathbf{M} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})\rangle, \mathcal{V}\rangle$ and an interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})$, the semantics of RPNL^- (resp., of RPNL^+) is defined recursively by the *satisfiability relation* $\Vdash$ as follows:

- for every propositional letter $p \in AP$, $\mathbf{M}, [d_i, d_j] \Vdash p$ iff $p \in \mathcal{V}([d_i, d_j])$;
- $\mathbf{M}, [d_i, d_j] \Vdash \neg\psi$ iff $\mathbf{M}, [d_i, d_j] \not\Vdash \psi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \psi_1 \vee \psi_2$ iff $\mathbf{M}, [d_i, d_j] \Vdash \psi_1$, or $\mathbf{M}, [d_i, d_j] \Vdash \psi_2$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle A\rangle\psi$ (resp., $\Diamond_r\psi$) iff $\exists d_k \in D$, $d_k > d_j$ (resp., $d_k \geq d_j$), such that $\mathbf{M}, [d_j, d_k] \Vdash \psi$.

We place ourselves in the most general setting and we do not impose any constraint on the valuation function. It particular, given interval $[d_i, d_j]$, it may happen that $p \in \mathcal{V}([d_i, d_j])$ and $p \notin \mathcal{V}([d'_i, d'_j])$ for all intervals $[d'_i, d'_j]$ (strictly) contained in $[d_i, d_j]$.

Let d_0 be the initial point of D and let d_1 be its successor. Since our logic has only future time operators, we can restrict our attention to the *initial interval* $[d_0, d_1]$ of $\mathbb{I}(\mathbb{D})$. From now on, we shall say that a formula φ is *satisfiable* if and only if there exists a model $\mathbf{M} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})\rangle, \mathcal{V}\rangle$ such that $\mathbf{M}, [d_0, d_1] \Vdash \varphi$, where $[d_0, d_1]$ is the initial interval of $\mathbb{I}(\mathbb{D})$.

3 An intuitive account of the proposed solution

In this section we give an intuitive account of the proposed solution to the satisfiability problem for RPNL^-. More precisely, we introduce the main features of a model building process that, given a formula φ to be checked for satisfiability, generates a model for it (if any) step by step. Such a process takes into consideration one element of the temporal domain at a time and, at each step, it progresses from one time point to the next one. For the moment, we completely ignore the problem of termination. In the following, we shall show how to turn this process into an effective procedure.

Let $D = \{d_0, d_1, d_2, \ldots\}$ be the temporal domain, which we assumed to be isomorphic to $\mathbb{N}$ or to a prefix of it. The model building process begins from time point d_1 by considering the initial interval $[d_0, d_1]$. It associates with $[d_0, d_1]$ the set $A_{[d_0,d_1]}$ of all and only the formulae which hold over it. Next, it moves from d_1 to its immediate successor d_2 and it takes into consideration the two intervals ending in d_2, namely, $[d_0, d_2]$ and $[d_1, d_2]$. As before, it associates with $[d_1, d_2]$ (resp. $[d_0, d_2]$) the set $A_{[d_1,d_2]}$ (resp. $A_{[d_0,d_2]}$) of all and only the formulae which hold over $[d_1, d_2]$ (resp. $[d_0, d_2]$). Since $[d_1, d_2]$ is a right neighbor of $[d_0, d_1]$, if $[A]\psi$ holds over $[d_0, d_1]$, then ψ must hold over $[d_1, d_2]$. Hence, for every formula $[A]\psi$ in $A_{[d_0,d_1]}$, it puts ψ in $A_{[d_1,d_2]}$. Moreover, since every interval which is a right neighbor of $[d_0, d_2]$ is also a right neighbor of $[d_1, d_2]$, and vice versa, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, ψ holds over $[d_0, d_2]$ if and only if it holds over $[d_1, d_2]$. Accordingly, it requires that $\psi \in A_{[d_0,d_2]}$ if and only if $\psi \in A_{[d_1,d_2]}$. Let us denote by $\mathrm{REQ}(d_2)$ the set of formulae of the form $\langle A\rangle\psi$ or $[A]\psi$ which hold over an interval ending in d_2 (by analogy, let $\mathrm{REQ}(d_1)$ be the set of formulae of the form $\langle A\rangle\psi$ or $[A]\psi$ which hold over an interval ending in d_1, that is, the formulae $\langle A\rangle\psi$ or $[A]\psi$ which hold over $[d_0, d_1]$). Next, the process moves from d_2 to its immediate successor d_3 and it takes into consideration the three intervals ending in d_3, namely, $[d_0, d_3]$, $[d_1, d_3]$, and $[d_2, d_3]$. As at the previous steps, for $i = 0, 1, 2$, it associates the set $A_{[d_i,d_3]}$ with $[d_i, d_3]$. Since $[d_1, d_3]$ is a right neighbor of $[d_0, d_1]$, for every formula $[A]\psi \in \mathrm{REQ}(d_1)$, $\psi \in A_{[d_1,d_3]}$. Moreover, $[d_2, d_3]$ is a right neighbor of both $[d_0, d_2]$ and $[d_1, d_2]$, and thus for every formula $[A]\psi \in \mathrm{REQ}(d_2)$, $\psi \in A_{[d_2,d_3]}$. Finally, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, we have that $\psi \in A_{[d_0,d_3]}$ if and only if $\psi \in A_{[d_1,d_3]}$ if and only if $\psi \in A_{[d_2,d_3]}$. Next, the process moves from d_3 to its successor d_4 and it repeats the same operations, and so on.

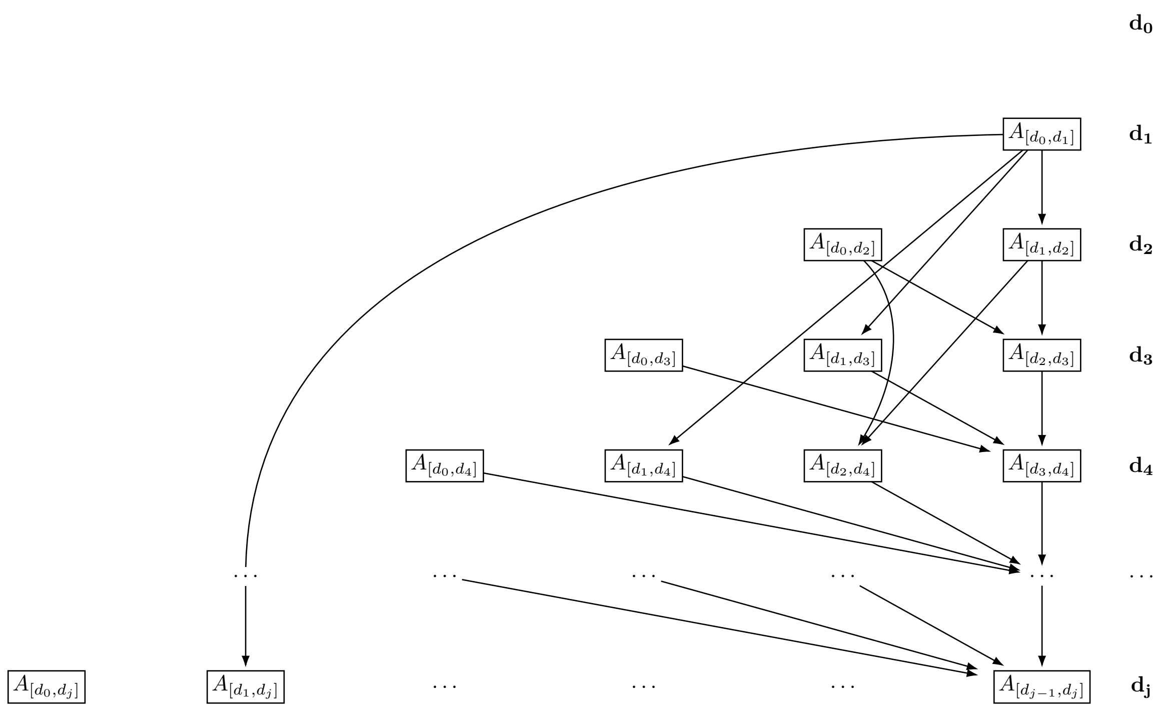


Fig. 3. The layered structure

The layered structure generated by the process is graphically depicted in Figure 3. The first layer corresponds to time point d_1, and for all $i > 1$, the i-th layer corresponds to time point d_i. If we associate with each node $A_{[d_i,d_j]}$ the corresponding interval $[d_i, d_j]$, we can interpret the set of edges as the neighborhood relation between pairs of intervals. As a general rule, given a time point $d_j \in D$, for every $d_i < d_j$, the set $A_{[d_i,d_j]}$ of all and only the formulae which hold over $[d_i, d_j]$ satisfies the following conditions:

- since $[d_i, d_j]$ is a right neighbor of every interval ending in d_i, for every formula $[A]\psi \in \text{REQ}(d_i)$, $\psi \in A_{[d_i,d_j]}$;
- since every right neighbor of $[d_i, d_j]$ is also a right neighbor of all intervals $[d_k, d_j]$ belonging to layer d_j, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, $\psi \in A_{[d_i,d_j]}$ if and only if it belongs to all sets $A_{[d_k,d_j]}$ belonging to the layer.

In [18], Bresolin and Montanari turn such a model building process into an effective tableau-based decision procedure for RPNL$^-$. Given an RPNL$^-$ formula φ, the procedure builds a tableau for φ whose (macro)nodes correspond to the layers of the structure in Figure 3 and whose edges connect pairs of nodes that correspond to consecutive layers. Unlike other tableau methods for interval temporal logics, where each node corresponds to a single interval [48,53], such a method associates any *set* of intervals $[d_i, d_j]$ ending at the same point d_j with a single node, whose label consists of a set of sets of formulae $A_{[d_i,d_j]}$ (one for every interval ending in d_j). Moreover, two nodes are connected by an edge (only) if their labels satisfy suitable constraints encoding the neighborhood relation among the associated intervals. Formulae devoid of temporal operators as well as formulae of the form $[A]\psi$ are satisfied by construction. Establishing the satisfiability of φ thus reduces to finding a (possibly infinite) path of nodes on which formulae of the form $\langle A\rangle\psi$ are satisfied as well (*fulfilling path*). To find such a path, the decision procedure first generates the whole (finite) tableau for φ; then it progressively removes parts of the tableau that cannot participate in a fulfilling path. It can be proved that φ is satisfiable if and only if the final tableau obtained by this pruning process is not empty.

As for the computational complexity, we have that the number of nodes of the tableau is $2^{2^{O(|\varphi|)}}$ and that, to determine the existence of a fulfilling path, the algorithm may take time polynomial in the number of nodes. Hence, the algorithm has a time complexity that is doubly exponential in the size of φ. Its performance can be improved by exploiting nondeterminism to guess a fulfilling path for the formula φ. In such a case, the fulfilling path can be built one node at a time: at each step, the procedure guesses the next node in the path and it moves from the current node to such a node. Since every (macro)node maintains the set of existential temporal formulae which have not been satisfied yet, at any time the algorithm basically needs to store only a pair of consecutive nodes in the path, namely, the current and the next ones, rather than the entire path. Hence, such a nondeterministic variant of the algorithm needs an amount of space which is exponential in the size of the formula, thus providing an EXPSPACE decision procedure for RPNL$^-$.

In the following, we shall develop an alternative NEXPTIME decision procedure for RPNL$^-$, interpreted over natural numbers, and we shall prove its optimality. Such a procedure follows the above-described approach, but its nodes are the single sets $A_{[d_i,d_j]}$, instead of layers, of the structure depicted in Figure 3. In such a way, the procedure avoids the double exponential blow-up of the method given in [18].

4 Labelled Interval Structures and Satisfiability

In this section we introduce some preliminary notions and we establish some basic results on which our tableau method for RPNL$^-$ relies.
Let φ be an RPNL$^-$ formula to be checked for satisfiability and let AP be the set of its propositional letters. For the sake of brevity, we use $(A)\psi$ as a shorthand for both $\langle A\rangle\psi$ and $[A]\psi$.

Definition 18. *The* closure $\mathrm{CL}(\varphi)$ *of* φ *is the set of all subformulae of* φ *and of their negations (we identify* $\neg\neg\psi$ *with* ψ*).*

Definition 19. *The set of* temporal requests *of* φ *is the set* $\mathrm{TF}(\varphi)$ *of all temporal formulae in* $\mathrm{CL}(\varphi)$*, that is,* $\mathrm{TF}(\varphi) = \{(A)\psi \in \mathrm{CL}(\varphi)\}$.

By induction on the structure of φ, we can easily prove the following proposition.

Proposition 8. *For every formula* φ*,* $|\mathrm{CL}(\varphi)|$ *is less than or equal to* $2 \cdot |\varphi|$*, while* $|\mathrm{TF}(\varphi)|$ *is less than or equal to* $2 \cdot (|\varphi| - 1)$.

The notion of φ-atom is defined in the standard way.

Definition 20. *A* φ-atom *is a set* $A \subseteq \mathrm{CL}(\varphi)$ *such that:*
- *for every* $\psi \in \mathrm{CL}(\varphi)$*,* $\psi \in A$ *iff* $\neg\psi \notin A$*;*
- *for every* $\psi_1 \vee \psi_2 \in \mathrm{CL}(\varphi)$*,* $\psi_1 \vee \psi_2 \in A$ *iff* $\psi_1 \in A$ *or* $\psi_2 \in A$.

We denote the set of all φ-atoms by A_φ. We have that $|A_\varphi| \leq 2^{|\varphi|}$. Atoms are connected by the following binary relation.

Definition 21. *Let* R_φ *be a binary relation over* A_φ *such that, for every pair of atoms* $A, A' \in A_\varphi$*,* $A \; R_\varphi \; A'$ *if and only if, for every* $[A]\psi \in \mathrm{CL}(\varphi)$*, if* $[A]\psi \in A$*, then* $\psi \in A'$.

We now introduce a suitable labelling of interval structures based on φ-atoms.

Definition 22. *A* φ-labelled interval structure *(*LIS *for short) is a pair* $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$*, where* $\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle$ *is an interval structure and* $\mathcal{L} : \mathbb{I}(\mathbb{D})^- \to A_\varphi$ *is a* labelling function *such that, for every pair of neighboring intervals* $[d_i, d_j], [d_j, d_k] \in \mathbb{I}(\mathbb{D})^-$*,* $\mathcal{L}([d_i, d_j]) \; R_\varphi \; \mathcal{L}([d_j, d_k])$.

If we interpret the labelling function as a valuation function, LISs represent *candidate models* for φ. The truth of formulae devoid of temporal operators and that of $[A]$-formulae indeed follow from the definition of φ-atom and the definition of R_φ, respectively. However, to obtain a model for φ we must also guarantee the truth of $\langle A\rangle$-formulae. To this end, we introduce the notion of fulfilling LIS.

Definition 23. *A* φ*-labelled interval structure* $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ *is* fulfilling *if and only if, for every temporal formula* $\langle A\rangle\psi \in \mathrm{TF}(\varphi)$ *and every interval* $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$*, if* $\langle A\rangle\psi \in \mathcal{L}([d_i, d_j])$*, then there exists* $d_k > d_j$ *such that* $\psi \in \mathcal{L}([d_j, d_k])$.

The following theorem proves that for any given formula φ, the satisfiability of φ is equivalent to the existence of a fulfilling LIS with the initial interval labelled by φ. The implication from left to right is straightforward; the opposite implication is proved by induction on the structure of the formula.

Theorem 29. *A formula φ is satisfiable if and only if there exists a fulfilling* LIS $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ *with* $\varphi \in \mathcal{L}([d_0, d_1])$.

Proof. Let φ be a satisfiable formula and let $\mathbf{M} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$ be a model for it. We define a LIS $\mathbf{L_M} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}_\mathbf{M}\rangle$ such that for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, $\mathcal{L}_\mathbf{M}([d_i, d_j]) = \{\psi \in \mathrm{CL}(\varphi) : \mathbf{M}, [d_i, d_j] \Vdash \psi\}$. It is immediate to prove that $\mathbf{L_M}$ is a fulfilling LIS and $\varphi \in \mathcal{L}_\mathbf{M}([d_0, d_1])$.
As for the opposite implication, let $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ be a fulfilling LIS with $\varphi \in \mathcal{L}([d_0, d_1])$. We define a model $\mathbf{M_L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}_\mathbf{L}\rangle$ such that for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$ and every propositional letter $p \in AP$, $p \in \mathcal{V}_\mathbf{L}([d_i, d_j])$ if and only if $p \in \mathcal{L}([d_i, d_j])$. We prove by induction on the structure of φ that for every $\psi \in \mathrm{CL}(\varphi)$ and every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, $\mathbf{M_L}, [d_i, d_j] \Vdash \psi$ if and only if $\psi \in \mathcal{L}([d_i, d_j])$. Since $\varphi \in \mathcal{L}([d_0, d_1])$, we can conclude that $\mathbf{M_L}, [d_0, d_1] \Vdash \varphi$.

- If ψ is the propositional letter p, then $p \in \mathcal{L}([d_i, d_j]) \overset{\mathcal{V}_\mathbf{L}\text{ def.}}{\Longleftrightarrow} p \in \mathcal{V}_\mathbf{L}([d_i, d_j]) \Longleftrightarrow \mathbf{M_L}, [d_i, d_j] \Vdash p$.
- If ψ is the formula $\neg\xi$, then $\neg\xi \in \mathcal{L}([d_i, d_j]) \overset{\text{atom def.}}{\Longleftrightarrow} \xi \notin \mathcal{L}([d_i, d_j]) \overset{\text{ind. hyp.}}{\Longleftrightarrow} \mathbf{M_L}, [d_i, d_j] \nVdash \xi \Longleftrightarrow \mathbf{M_L}, [d_i, d_j] \Vdash \neg\xi$.
- If ψ is the formula $\xi_1 \vee \xi_2$, then $\xi_1 \vee \xi_2 \in \mathcal{L}([d_i, d_j]) \overset{\text{atom def.}}{\Longleftrightarrow} \xi_1 \in \mathcal{L}([d_i, d_j])$ or $\xi_2 \in \mathcal{L}([d_i, d_j]) \overset{\text{ind. hyp.}}{\Longleftrightarrow} \mathbf{M_L}, [d_i, d_j] \Vdash \xi_1$ or $\mathbf{M_L}, [d_i, d_j] \Vdash \xi_2 \Longleftrightarrow \mathbf{M_L}, [d_i, d_j] \Vdash \xi_1 \vee \xi_2$.
- Let ψ be the formula $\langle A\rangle\xi$. Suppose that $\langle A\rangle\xi \in \mathcal{L}([d_i, d_j])$. Since $\mathbf{L}$ is fulfilling, there exists an interval $[d_j, d_k] \in \mathbb{I}(\mathbb{D})^-$ such that $\xi \in \mathcal{L}([d_j, d_k])$. By inductive hypothesis, we have that $\mathbf{M_L}, [d_j, d_k] \Vdash \xi$, and hence $\mathbf{M_L}, [d_i, d_j] \Vdash \langle A\rangle\xi$. As for the opposite implication, assume by contradiction that $\mathbf{M_L}, [d_i, d_j] \Vdash \langle A\rangle\xi$, but $\langle A\rangle\xi \notin \mathcal{L}([d_i, d_j])$. By atom definition, this implies that $\neg\langle A\rangle\xi = [A]\neg\xi \in \mathcal{L}([d_i, d_j])$. By definition of LIS, we have that, for every $d_k > d_j$, $\mathcal{L}([d_i, d_j]) \; R_\varphi \; \mathcal{L}([d_j, d_k])$, and thus $\neg\xi \in \mathcal{L}([d_j, d_k])$. By inductive hypothesis, this implies that $\mathbf{M_L}, [d_j, d_k] \Vdash \neg\xi$ for every $d_k > d_j$, and hence $\mathbf{M_L}, [d_i, d_j] \Vdash [A]\neg\xi$, which contradicts the hypothesis that $\mathbf{M_L}, [d_i, d_j] \Vdash \langle A\rangle\xi$. □

Theorem 29 reduces the satisfiability problem for φ to the problem of finding a fulfilling LIS with the initial interval labelled by φ. From now on, we say that a fulfilling LIS $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ *satisfies* φ if and only if $\varphi \in \mathcal{L}([d_0, d_1])$.
Since fulfilling LISs satisfying φ may be arbitrarily large or even infinite, we must find a way to finitely establish their existence. In the following, we first give a bound on the size of finite fulfilling LISs that must be checked for satisfiability, when searching for finite φ-models; then, we show that we can restrict ourselves to infinite fulfilling LISs with a finite bounded representation, when searching for infinite φ-models. To prove these results, we take advantage of the following two fundamental properties of LISs:

1. *The labellings of a pair of intervals $[d_i, d_j], [d_k, d_j]$ with the same right endpoint must agree on temporal formulae.*
   Since every right neighbor of $[d_i, d_j]$ is also a right neighbor of $[d_k, d_j]$, we have that for every existential formula $\langle A\rangle\psi \in \mathrm{TF}(\varphi)$, $\langle A\rangle\psi \in \mathcal{L}([d_i, d_j])$ iff $\langle A\rangle\psi \in \mathcal{L}([d_k, d_j])$ (it easily follows from Definitions 20, 21, and 22). The same holds for universal formulae $[A]\psi$.
2. *$\frac{|\mathrm{TF}(\varphi)|}{2}$ right neighboring intervals suffice to fulfill the existential formulae belonging to the labelling of an interval $[d_i, d_j]$.*
   The number of right neighboring intervals which are needed to fulfill all existential formulae of $\mathcal{L}([d_i, d_j])$ is bounded by the number of $\langle A\rangle$-formulae in $\mathrm{TF}(\varphi)$, which is

equal to $\frac{|\mathrm{TF}(\varphi)|}{2}$ (in the worst case, different existential formulae are satisfied by different right neighboring intervals).

Definition 24. *Given a* LIS $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ *and* $d \in D$*, we denote by* $\mathrm{REQ}(d)$ *the set of all and only the temporal formulae belonging to the labellings of the intervals ending in* d*.*

We denote by REQ_φ the set of all possible sets of requests. It is not difficult to show that $|\mathrm{REQ}_\varphi|$ is equal to $2^{\frac{\mathrm{TF}(\varphi)}{2}}$.

Definition 25. *Given a* LIS $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$*, a set of points* $D' \subseteq D$*, and a set of temporal formulae* $\mathcal{R} \subseteq \mathrm{TF}(\varphi)$*, we say that* $\mathcal{R}$ occurs n times in D' *if and only if there exist exactly* n *distinct points* $d_{i_1}, \ldots, d_{i_n} \in D'$ *such that* $\mathrm{REQ}(d_{i_j}) = \mathcal{R}$*, for all* $1 \leq j \leq n$*.*

Theorem 30. *Let* $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ *be a* finite *fulfilling* LIS *that satisfies* φ *and let* $m = \frac{|\mathrm{TF}(\varphi)|}{2}$*. Then there exists a* finite *fulfilling* LIS $\overline{\mathbf{L}} = \langle\langle \overline{\mathbb{D}}, \mathbb{I}(\overline{\mathbb{D}})^- \rangle, \overline{\mathcal{L}}\rangle$ *that satisfies* φ *such that, for every* $\overline{d}_i \in \overline{D}$*,* $\mathrm{REQ}(\overline{d}_i)$ *occurs at most* m *times in* $\overline{D} \setminus \{d_1\}$*.*

Proof. Let $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ be a finite fulfilling LIS that satisfies φ. If for every $d_i \in D$, $\mathrm{REQ}(d_i)$ occurs at most m times in $D \setminus \{d_1\}$, we are done. If this is not the case, we show how to build a fulfilling LIS with the requested property by progressively removing exceeding points from D.
Let $\mathbf{L}_0 = \mathbf{L}$ and let $\mathcal{R}_0 = \{\mathrm{REQ}_1, \mathrm{REQ}_2, \ldots, \mathrm{REQ}_k\}$ be the (arbitrarily ordered) finite set of all and only the sets of requests that occur more than m times in $D \setminus \{d_1\}$. We show how to turn $\mathbf{L}_0$ into a fulfilling LIS $\mathbf{L}_1 = \langle\langle \mathbb{D}_1, \mathbb{I}(\mathbb{D}_1)^- \rangle, \mathcal{L}_1\rangle$ satisfying φ, which, unlike $\mathbf{L}_0$, contains exactly m points $d \in D_1 \setminus \{d_1\}$ such that $\mathrm{REQ}(d) = \mathrm{REQ}_1$. By iterating such a transformation $k-1$ times, we can turn $\mathbf{L}_1$ into a fulfilling LIS devoid of exceeding points that satisfies φ.
The fulfilling LIS $\mathbf{L}_1 = \langle\langle \mathbb{D}_1, \mathbb{I}(\mathbb{D}_1)^- \rangle, \mathcal{L}_1\rangle$ can be obtained as follows. Let $d_{j_1}, d_{j_2}, \ldots, d_{j_n}$, with $d_1 < d_{j_1} < d_{j_2} < \ldots < d_{j_n}$, be the n points in D, with $n > m$, such that $\mathrm{REQ}(d_{j_i}) = \mathrm{REQ}_1$. We define $\mathbb{D}'_1 = \langle D \setminus \{d_{j_1}\}, < \rangle$ and $\mathcal{L}'_1 = \mathcal{L}|_{\mathbb{I}(\mathbb{D}'_1)^-}$ (the restriction of $\mathcal{L}$ to the intervals on $\mathbb{D}'_1$). The pair $\mathbf{L}'_1 = \langle\langle \mathbb{D}'_1, \mathbb{I}(\mathbb{D}'_1)^- \rangle, \mathcal{L}'_1\rangle$ is obviously a finite LIS, but it is not necessarily a fulfilling one. The removal of d_{j_1} causes the removal of all intervals either beginning or ending at d_{j_1}. While the removal of intervals beginning at d_{j_1} is not critical (intervals ending at d_{j_1} are removed as well), there can be some points $d < d_{j_1}$ such that some formulae $\langle A\rangle\psi \in \mathrm{REQ}(d)$ are fulfilled in $\mathbf{L}_0$, but they are not fulfilled in $\mathbf{L}'_1$ anymore. We fix such defects (if any) one-by-one by properly redefining $\mathcal{L}'_1$. Let $d < d_{j_1}$ and $\langle A\rangle\psi \in \mathrm{REQ}(d)$ such that $\psi \in \mathcal{L}([d, d_{j_1}])$ and there exists no $d' \in D \setminus \{d_{j_1}\}$ such that $\psi \in \mathcal{L}'_1([d, d'])$. Since $\mathrm{REQ}(d)$ contains at most m $\langle A\rangle$-formulae, there exists at least one point $d_{j_i} \in \{d_{j_2}, \ldots, d_{j_n}\}$ such that the atom $\mathcal{L}'_1([d, d_{j_i}])$ either fulfills no $\langle A\rangle$-formulae or it fulfills only $\langle A\rangle$-formulae which are also fulfilled by some other φ-atom $\mathcal{L}'_1([d, d'])$. Let d_{j_i} one of such "useless" points. We can redefine $\mathcal{L}'_1([d, d_{j_i}])$ by putting $\mathcal{L}'_1([d, d_{j_i}]) = \mathcal{L}([d, d_{j_1}])$, thus fixing the problem with $\langle A\rangle\psi \in \mathrm{REQ}(d)$. Notice that, since $\mathrm{REQ}(d_{j_1}) = \mathrm{REQ}(d_{j_i}) = \mathrm{REQ}_1$, such a change has no impact on the right neighboring intervals of $[d, d_{j_i}]$. In a similar way, we can fix the other possible defects caused by the removal of d_{j_1}. We repeat such a process until we remain with exactly m distinct points d such that $\mathrm{REQ}(d) = \mathrm{REQ}_1$. Let $\mathbf{L}_1 = \langle\langle \mathbb{D}_1, \mathbb{I}(\mathbb{D}_1)\rangle, \mathcal{L}_1\rangle$ be the resulting LIS. It is immediate to show that it is fulfilling and that it satisfies φ. □

To deal with the case of infinite (fulfilling) LISs, we introduce the notion of *ultimately periodic* LISs.

Definition 26. *An infinite* LIS $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ *is* ultimately periodic*, with prefix* l *and period* $p > 0$*, if and only if for all* $i > l$, $\mathrm{REQ}(d_i) = \mathrm{REQ}(d_{i+p})$.

The following theorem shows that if there exists an infinite fulfilling LIS that satisfies φ, then there exists an ultimately periodic fulfilling one that satisfies φ. Furthermore, it provides a bound to the prefix and period of such a fulfilling LIS which closely resembles the one that we established for finite fulfilling LISs.

Theorem 31. *Let* $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ *be an* infinite *fulfilling* LIS *that satisfies* φ *and let* $m = \frac{|\mathrm{TF}(\varphi)|}{2}$. *Then there exists an ultimately periodic fulfilling* LIS $\overline{\mathbf{L}} = \langle\langle \overline{\mathbb{D}}, \mathbb{I}(\overline{\mathbb{D}})^- \rangle, \overline{\mathcal{L}}\rangle$, *with prefix* l *and period* p*, that satisfies* φ *such that:*

1. *for every pair of points* $\overline{d}_i, \overline{d}_j \in \overline{D}$*, with* $\overline{d}_0 \leq \overline{d}_i \leq \overline{d}_l$ *and* $\overline{d}_j > \overline{d}_l$, $\mathrm{REQ}(\overline{d}_i) \neq \mathrm{REQ}(\overline{d}_j)$*, that is, points belonging to the prefix and points belonging to the period have different sets of requests;*
2. *for every* $\overline{d}_i \in \overline{D}$*, with* $\overline{d}_0 \leq \overline{d}_i \leq \overline{d}_l$, $\mathrm{REQ}(\overline{d}_i)$ *occurs at most* m *times in* $\{\overline{d}_2, \ldots, \overline{d}_l\}$*;*
3. *for every pair of points* $\overline{d}_i, \overline{d}_j \in \overline{D}$*, with* $\overline{d}_{l+1} \leq \overline{d}_i, \overline{d}_j \leq \overline{d}_{l+p}$*, if* $i \neq j$*, then* $\mathrm{REQ}(\overline{d}_i) \neq \mathrm{REQ}(\overline{d}_j)$.

Proof. Let φ be a satisfiable formula and let $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}\rangle$ be an infinite fulfilling LIS that satisfies φ. We define the following sets:

- $Fin(\mathbf{L}) = \{\mathrm{REQ}(d_i)$: there exists a *finite* number of points $d \in \mathbb{D}$ such that $\mathrm{REQ}(d) = \mathrm{REQ}(d_i)\}$;
- $Inf(\mathbf{L}) = \{\mathrm{REQ}(d_i)$: there exists an *infinite* number of points $d \in \mathbb{D}$ such that $\mathrm{REQ}(d) = \mathrm{REQ}(d_i)\}$.

We build an infinite ultimately periodic LIS $\overline{\mathbf{L}}$, with prefix $l \leq m \cdot |Fin(\mathbf{L})| + 1$ and period $p = |Inf(\mathbf{L})|$, that satisfies φ as follows.
Let $n \in \mathbb{N}$ be the index of the smallest point in $\mathbb{D}$ such that, for every $i \geq n$, $\mathrm{REQ}(d_i) \in Inf(\mathbf{L})$. We first collect and (arbitrarily) enumerate the elements of $Inf(\mathbf{L})$, that is, let $Inf(\mathbf{L}) = \{\mathrm{REQ}_0, \ldots, \mathrm{REQ}_{p-1}\}$. The cardinality p of $Inf(\mathbf{L})$ is the period of $\overline{\mathbf{L}}$. Next, we define an ultimately periodic LIS $\mathbf{L}' = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{L}'\rangle$, with prefix $n-1$ and period p, in such a way that:

- for all $0 \leq i < n$, $\mathrm{REQ}(d_i)$ remains unchanged;
- for all $i \geq n$, $\mathrm{REQ}(d_i)$ becomes $\mathrm{REQ}_{(i-n) \bmod p}$.

To make the ultimately periodic LIS $\mathbf{L}'$ a fulfilling LIS satisfying φ which respects the above conditions on $\mathrm{REQ}(d_i)$, for all $i \geq 0$, we must properly define the labelling $\mathcal{L}'$. We distinguish two cases:

(i) $d_1 \leq d_i < d_n$. Since $\mathbf{L}$ is a fulfilling LIS, for all $\langle A\rangle\psi \in \mathrm{REQ}(d_i)$ there exists a point $d_j > d_i$ such that $\psi \in \mathcal{L}([d_i, d_j])$. Two cases may arise:
   - if $d_j < d_n$, we put $\mathcal{L}'([d_i, d_j]) = \mathcal{L}([d_i, d_j])$;
   - if $d_j \geq d_n$, then $\mathrm{REQ}(d_j)$ in $\mathbf{L}$ belongs to $Inf(\mathbf{L})$. We have that there exist infinitely many points $d_k \geq d_n$ such that $\mathrm{REQ}(d_k)$ in $\mathbf{L}'$ is equal to $\mathrm{REQ}(d_j)$ in $\mathbf{L}$. Let d_k be one of such points for which the labelling $\mathcal{L}'([d_i, d_k])$ has not been defined yet. We fulfill $\langle A\rangle\psi$ in $\mathbf{L}'$ by putting $\mathcal{L}'([d_i, d_k]) = \mathcal{L}([d_i, d_j])$.

   We repeat such a construction until we fulfill (in $\mathbf{L}'$) all $\langle A\rangle$-formulae belonging to $\mathrm{REQ}(d_i)$. To complete the labelling $\mathcal{L}'$ on the remaining intervals $[d_i, d_k]$, we (arbitrarily) put $\mathcal{L}'([d_i, d_k]) = \mathcal{L}([d_i, d_j])$, with d_j such that $\mathrm{REQ}(d_k)$ in $\mathbf{L}'$ is equal to $\mathrm{REQ}(d_j)$ in $\mathbf{L}$.

(ii) $d_i \geq d_n$. We have that there exists a point $d_j \geq d_n$ such that $\mathrm{REQ}(d_j)$ in $\mathbf{L}$ is equal to $\mathrm{REQ}(d_i)$ in $\mathbf{L}'$. Let $\langle A\rangle\psi \in \mathrm{REQ}(d_i)$ (in $\mathbf{L}'$). Since $\mathbf{L}$ is fulfilling, there exists a point

$d_k > d_j$, with $\mathrm{REQ}(d_k) \in Inf(\mathbf{L})$, such that $\psi \in \mathcal{L}([d_j, d_k])$. From $\mathrm{REQ}(d_k) \in Inf(\mathbf{L})$, it follows that there exist infinitely many points $d_l > d_i$ such that $\mathrm{REQ}(d_l)$ in $\mathbf{L}'$ is equal to $\mathrm{REQ}(d_k)$ in $\mathbf{L}$. Let d_l be one of such points for which the labelling $\mathcal{L}'([d_i, d_l])$ has not been defined yet. We fulfill $\langle A\rangle\psi$ by putting $\mathcal{L}'([d_i, d_l]) = \mathcal{L}([d_j, d_k])$. We can repeat such a construction until we fulfill (in $\mathbf{L}'$) all $\langle A\rangle$-formulae belonging to $\mathrm{REQ}(d_i)$. To complete the labelling of the remaining intervals $[d_i, d_l]$, we arbitrarily put $\mathcal{L}'([d_i, d_l]) = \mathcal{L}([d_j, d_k])$, with d_k such that $\mathrm{REQ}(d_k)$ in $\mathbf{L}$ is equal to $\mathrm{REQ}(d_l)$ in $\mathbf{L}'$.

Condition 3 of the theorem is respected by construction. Conditions 1 and 2 require that the prefix $\{d_0, \ldots, d_{n-1}\}$ does not include points d such that $\mathrm{REQ}(d) \in Inf(\mathbf{L})$ and that for every point d in $\{d_0, \ldots, d_{n-1}\}$, $\mathrm{REQ}(d)$ occurs at most m times in $\{d_2, \ldots, d_{n-1}\}$, respectively. These two conditions are not necessarily guaranteed by $\mathbf{L}'$. We turn $\mathbf{L}'$ into a fulfilling ultimately periodic LIS $\overline{\mathbf{L}}$ satisfying φ that respects conditions 1 and 2 by means of a two-step removal process.

- **Step 1.** We replace the LIS $\mathbf{L}'$ by a LIS $\mathbf{L}'' = \langle\langle \mathbb{D}'', \mathbb{I}(\mathbb{D}'')^-\rangle, \mathcal{L}''\rangle$ which is obtained from $\mathbf{L}'$ by deleting all points $d_1 < d_i < d_n$ such that $\mathrm{REQ}(d_i) \in Inf(\mathbf{L})$. The resulting LIS $\mathbf{L}''$ is still ultimately periodic, but it is not necessarily fulfilling. As an effect of the removal of d_i, there can be some point d in the prefix such that some formula $\langle A\rangle\psi \in \mathrm{REQ}(d)$ was fulfilled in $\mathcal{L}'$, but it is not fulfilled in $\mathcal{L}''$ anymore. We can fix such a defect by redefining $\mathcal{L}''$ in an appropriate way. Since $\mathbf{L}'$ is fulfilling, there exists a point $d_j > d$ such that $\psi \in \mathcal{L}'([d, d_j])$ and d_j does not belong to $\mathbf{L}''$. This happens if (and only if) $\mathrm{REQ}(d_j) \in Inf(\mathbf{L})$ and thus there exist infinitely many points d_k such that $\mathrm{REQ}(d_k)$ in $\mathbf{L}''$ is equal to $\mathrm{REQ}(d_j)$ in $\mathbf{L}'$. Let d_k be one of these points which is "useless" (see the proof of Theorem 30). We can fulfill $\langle A\rangle\psi$ by putting $\mathcal{L}''([d, d_k]) = \mathcal{L}'([d, d_j])$. In a similar way, we can fix the other possible defects caused by the removal of d_i. By repeating such a process for every other point that must be removed (if any), we guarantee that $\mathbf{L}''$ is a fulfilling ultimately periodic LIS that satisfies φ.
- **Step 2.** If there exists some point d'' belonging to the prefix $\{d''_0, \ldots, d''_h\}$ of $\mathbf{L}''$ such that $\mathrm{REQ}(d'')$ occurs more than m times in $\{d''_2, \ldots, d''_h\}$, we can proceed as in the proof of Theorem 30 to obtain a new fulfilling ultimately periodic LIS $\overline{\mathbf{L}}$ satisfying φ with prefix $\{\overline{d}_0, \ldots, \overline{d}_l\}$ such that every set of requests of the prefix is repeated at most m times in $\{\overline{d}_2, \ldots, \overline{d}_l\}$. □

5 The complexity of the satisfiability problem for RPNL$^-$

In this section we provide a precise characterization of the computational complexity of the satisfiability problem for RPNL$^-$.

5.1 An upper bound to the computational complexity

A decision procedure for RPNL$^-$ can be derived from the results of Section 4 in a straightforward way. Theorems 30 and 31 indeed provide a bound on the size of the LISs to be checked:

- by Theorem 30, we have that if there exists a finite LIS satisfying φ, then there exists a finite one of size less than or equal to $|\mathrm{REQ}_\varphi| \cdot \frac{|\mathrm{TF}(\varphi)|}{2} + 1$ which satisfies φ;
- by Theorem 31, we have that if there exists an infinite LIS satisfying φ, then there exists an ultimately periodic one, with prefix $l \leq |\mathrm{REQ}_\varphi| \cdot \frac{|\mathrm{TF}(\varphi)|}{2} + 1$ and period $p \leq |\mathrm{REQ}_\varphi|$, which satisfies φ.

A simple decision algorithm to check the satisfiability of an RPNL^- formula φ that nondeterministically guesses a LIS satisfying it can be defined as follows.

First, the algorithm guesses the set $\text{REQ}_{inf} = \{\text{REQ}_1, \ldots, \text{REQ}_p\}$, with $\text{REQ}_{inf} \subseteq \text{REQ}_\varphi$, of the requests that occur in the period. If $p = 0$, then it guesses the length $l \leq |\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1$ of a finite LIS. Otherwise, it takes p as the period of an ultimately periodic LIS and it guesses the length $l \leq (|\text{REQ}_\varphi| - p) \cdot \frac{|\text{TF}(\varphi)|}{2} + 1$ of its prefix. Next, the algorithm guesses the labelling of the initial interval $[d_0, d_1]$, taking an atom $A_{[d_0,d_1]}$ that includes φ, and it initializes a counter c to 1. If $c < l$, then it guesses the labelling of the intervals ending in d_2, that is, it associates an atom $A_{[d_0,d_2]}$ with $[d_0, d_2]$ and an atom $A_{[d_1,d_2]}$ with $[d_1, d_2]$ such that $A_{[d_0,d_1]}\ R_\varphi\ A_{[d_1,d_2]}$, with $\text{REQ}(d_2) \in \text{REQ}_\varphi \setminus \text{REQ}_{inf}$, and it increments c by one. The algorithm proceeds in this way, incrementing c by one for every point d_j it considers and checking that, for every pair of atoms $A_{[d_k,d_i]}$ and $A_{[d_i,d_j]}$, $A_{[d_k,d_i]}\ R_\varphi\ A_{[d_i,d_j]}$. For each point d_j, it must guarantee that $\text{REQ}(d_j) \in \text{REQ}_\varphi \setminus \text{REQ}_{inf}$ and that $\text{REQ}(d_j)$ occurs at most $\frac{|\text{TF}(\varphi)|}{2}$ times in $\{d_2, \ldots, d_j\}$.

When c reaches the value l, two cases are possible. If $p = 0$, then d_l is the last point of a finite LIS, and the algorithm checks whether it is fulfilling. If $p > 0$, it checks if the guessed prefix and period represent a fulfilling LIS by proceeding as follows:

- for every atom $A_{[d_i,d_j]}$ in the prefix and for every formula $\langle A \rangle \psi \in A_{[d_i,d_j]}$, it checks if either there exists an atom $A_{[d_j,d_k]}$ in the prefix that contains ψ or there exists an atom A' containing ψ and a set $\text{REQ}_h \in \text{REQ}_{inf}$ such that $\text{REQ}_h = A' \cap \text{TF}(\varphi)$ and $A_{[d_i,d_j]}\ R_\varphi\ A'$;
- for every atom $A_{[d_i,d_j]}$ in the prefix and for every set $\text{REQ}_h \in \text{REQ}_{inf}$, it checks if there exists an atom A' such that $\text{REQ}_h = A' \cap \text{TF}(\varphi)$ and $A_{[d_i,d_j]}\ R_\varphi\ A'$;
- for every set $\text{REQ}_h \in \text{REQ}_{inf}$ and for every formula $\langle A \rangle \psi \in \text{REQ}_h$, it checks if there exists an atom A' containing ψ and a set $\text{REQ}_k \in \text{REQ}_{inf}$ such that $\text{REQ}_k = A' \cap \text{TF}(\varphi)$ and $\text{REQ}_h\ R_\varphi\ A'$;
- for every pair of sets $\text{REQ}_h, \text{REQ}_k \in \text{REQ}_{inf}$, it checks if there exists an atom A' such that $\text{REQ}_k = A' \cap \text{TF}(\varphi)$ and $\text{REQ}_h\ R_\varphi\ A'$.

By Theorems 30 and 31, it follows that the algorithm returns *true* if and only if φ is satisfiable. As for the computational complexity of the algorithm, we observe that:

1. l is less than or equal to $|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1$, while p is less than or equal to $|\text{REQ}_\varphi|$;
2. for every point $d_1 \leq d_j \leq d_l$, the algorithm guesses exactly j atoms $A_{[d_i,d_j]}$;
3. checking for the fulfillness of the guessed LIS takes time polynomial in p and in the number of guessed atoms;
4. $|\text{TF}(\varphi)|$ is linear in the length of φ, while $|REQ_\varphi|$ is exponential in it.

Hence, if $|\varphi| = n$, the number of guessed sets in REQ_{inf} plus number of guessed atoms in the prefix is bounded by

$$
\begin{aligned}
|\text{REQ}_\varphi| + \sum_{i=1}^{|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1} i &= O\left(|REQ_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2}\right)^2 \\
&= \left(2^{O(n)} \cdot O(n)\right)^2 \\
&= 2^{2 \cdot O(n)} \cdot O(n^2) \\
&= 2^{O(n)} \cdot O(n^2) \\
&= 2^{O(n)},
\end{aligned}
$$

that is, it is exponential in the length of φ. This implies that the satisfiability problem for RPNL$^-$ can be solved by the above nondeterministic algorithm in nondeterministic exponential time.

Theorem 32. *The satisfiability problem for* RPNL$^-$*, over natural numbers, is in NEXPTIME.*

5.2 A lower bound to the computational complexity

We now provide a NEXPTIME lower bound for the complexity of the satisfiability problem for RPNL$^-$ by reducing to it the exponential tiling problem, which is known to be NEXPTIME-complete [8].
Let us denote by $\mathbb{N}_m$ the set of natural numbers less than m and by $N(m)$ the grid $\mathbb{N}_m \times \mathbb{N}_m$. A *domino system* is a triple $\mathcal{D} = \langle C, H, V \rangle$, where C is a finite set of *colors* and $H, V \subseteq C \times C$ are the horizontal and vertical *adjacency relations*. We say that $\mathcal{D}$ *tiles* $N(m)$ if there exists a mapping $\tau : N(m) \to C$ such that, for all $(x, y) \in N(m)$:
1. if $\tau(x, y) = c$ and $\tau(x + 1, y) = c'$, then $(c, c') \in H$;
2. if $\tau(x, y) = c$ and $\tau(x, y + 1) = c'$, then $(c, c') \in V$.

The exponential tiling problem consists in determining, given a natural number n and a domino system $\mathcal{D}$, whether $\mathcal{D}$ tiles $N(2^n)$ or not. Proving that the satisfiability problem for RPNL$^-$ is NEXPTIME-hard can be done by encoding the exponential tiling problem with a formula $\varphi(\mathcal{D})$, of length polynomial in n, which uses propositional letters to represent positions in the grid and colors, and by showing that $\varphi(\mathcal{D})$ is satisfiable if and only if $\mathcal{D}$ tiles $N(2^n)$. Such a formula consists of three main parts. The first part imposes a sort of locality principle; the second part encodes the $N(2^n)$ grid; the third part imposes that every point of the grid is tiled by exactly one color and that the colors respect the adjacency conditions.

Theorem 33. *The satisfiability problem for* RPNL$^-$*, over natural numbers, is NEXPTIME-hard.*

Proof. Given a domino system $\mathcal{D} = \langle C, H, V \rangle$, we build an RPNL$^-$ formula φ, of length polynomial in n, that is satisfiable if and only if $\mathcal{D}$ tiles $N(2^n)$.
The models for φ encode a tiling $\tau : N(2^n) \to C$ in the following way. First, we associate with every point $z = (x, y) \in N(2^n)$ a $2n$-bit word $(z_{2n-1} z_{2n-2} \ldots z_1 z_0) \in \{0, 1\}^{2n}$ such that $x = \sum_{i=0}^{n-1} z_i 2^i$ and $y = \sum_{i=n}^{2n-1} y_i 2^{i-n}$. Pairs of points $[z, t]$ of $N(2^n)$ are represented as intervals by means of the propositional letters Z_i, T_i, with $0 \leq i \leq 2n - 1$, as follows:

$$Z_i : z_i = 1; T_i : t_i = 1.$$

Moreover, the colors of $z = (x, y)$ and $t = (x', y')$ are expressed by means of the propositional letters Z_c, T_c, with $c \in C$, as follows:

$$Z_c : \tau(x, y) = c; T_c : \tau(x', y') = c.$$

To ease the writing of the formula φ encoding the tiling problem, we use the auxiliary propositional letters Z_i^* (for $0 \leq i \leq 2n - 1$) and ZH_i^* (for $n \leq i \leq 2n - 1$), with the following intended meaning:

$$Z_i^* : \text{ for all } 0 \leq j < i, z_j = 1; ZH_i^* : \text{ for all } n \leq j < i, z_j = 1.$$

To properly encode the tiling problem, we must constrain the relationships among these propositional letters.

Definition of auxiliary propositional letters. As a preliminary step, we define the auxiliary propositional letters Z_i^*, with $0 \le i \le 2n-1$, and ZH_i^*, with $n \le i \le 2n-1$, as follows:

$$[A][A]\left(Z_0^* \wedge \bigwedge_{i=1}^{2n-1} (Z_i^* \iff (Z_{i-1}^* \wedge Z_{i-1}))\right)$$

$$[A][A]\left(ZH_n^* \wedge \bigwedge_{i=n+1}^{2n-1} (ZH_i^* \iff (ZH_{i-1}^* \wedge Z_{i-1}))\right).$$

Let us call α the conjunction of the above two formulae.

Locality conditions. Then, we impose a sort of "locality principle" on the interpretation of the propositional letters. Given an interval $[z, t]$, we encode the position $z = (x, y)$ (resp., $t = (x', y')$) and its color $\tau(x, y)$ (resp., $\tau(x', y')$) by means of the propositional letters Z_i, Z_i^*, ZH_i^*, and Z_c (resp., T_i and T_c) by imposing the following constraints:

- all intervals $[z, w]$ starting in z must agree on the truth value of Z_i, Z_i^*, ZH_i^*, and Z_c;
- for every pair of neighboring intervals $[z, t], [t, w]$, the truth value of T_i and T_c over $[z, t]$ must agree with the truth value of Z_i and Z_c over $[t, w]$.

From the above constraints, it easily follows that all intervals $[w, t]$ ending in t must agree on the truth value of T_i and T_c.

Such constraints are encoded by the conjunction of the following formulae (let us call it β):

$$\bigwedge_{i=0}^{2n-1} (\langle A\rangle Z_i \Longrightarrow [A]Z_i) \wedge \bigwedge_{i=0}^{2n-1} [A](\langle A\rangle Z_i \Longrightarrow [A]Z_i)$$

$$\bigwedge_{i=0}^{2n-1} (\langle A\rangle Z_i^* \Longrightarrow [A]Z_i^*) \wedge \bigwedge_{i=0}^{2n-1} [A](\langle A\rangle Z_i^* \Longrightarrow [A]Z_i^*)$$

$$\bigwedge_{i=n}^{2n-1} (\langle A\rangle ZH_i^* \Longrightarrow [A]ZH_i^*) \wedge \bigwedge_{i=n}^{2n-1} [A](\langle A\rangle ZH_i^* \Longrightarrow [A]ZH_i^*)$$

$$\bigwedge_{c\in C} (\langle A\rangle Z_c \Longrightarrow [A]Z_c) \wedge \bigwedge_{c\in C} [A](\langle A\rangle Z_c \Longrightarrow [A]Z_c)$$

$$\bigwedge_{i=0}^{2n-1} [A](T_i \iff [A]Z_i) \wedge \bigwedge_{i=0}^{2n-1} [A][A](T_i \iff [A]Z_i)$$

$$\bigwedge_{c\in C} [A](T_c \iff [A]Z_c) \wedge \bigwedge_{c\in C} [A][A](T_c \iff [A]Z_c)$$

Encoding of the grid. Next, we must guarantee that every point $z = (x, y) \in N(2^n)$, with the exception of the upper-right corner $(2^n - 1, 2^n - 1)$, has a "successor" $t = (x', y')$, that is, if $x \neq 2^n - 1$, then $(x', y') = (x + 1, y)$; otherwise $(x = 2^n - 1)$, $(x', y') = (0, y + 1)$. Note that, thanks to our encoding of z and t, the binary encoding of the successor of z is equal to the binary encoding of z incremented by 1. Such a successor relation can be encoded as follows. Given two $2n$-bit words $z = \sum_{i=0}^{2n-1} z_i 2^i$ and $t = \sum_{i=0}^{2n-1} t_i 2^i$, we have that $t = z + 1$ if and only if there exists some $0 \le j \le 2n - 1$ such that:

1. $z_j = 0$ and, for all $i < j$, $z_i = 1$;
2. $t_j = 1$ and, for all $i < j$, $t_i = 0$;
3. for all $j < k \leq 2n-1$, $z_k = t_k$.

It is easy to show that, for every i, with $0 \leq i \leq 2n-1$, we can write t_i as $z_i \oplus \bigwedge_{k<i} z_k$, where $\oplus$ denotes the exclusive or. Taking advantage of this fact, the successor relation can be expressed by the following formula (let us call it γ):

$$[A]\left(\langle A\rangle\neg(Z^*_{2n-1} \wedge Z_{2n-1}) \Longrightarrow \langle A\rangle \bigwedge_{i=0}^{2n-1} (T_i \iff (Z_i \oplus Z^*_i))\right).$$

Furthermore, the left conjunct of the following formula (let us call it δ) encodes the initial point $(0,0)$ of the grid, while the right one encodes the final point $(2^n-1, 2^n-1)$:

$$\langle A\rangle\langle A\rangle \bigwedge_{i=0}^{2n-1} \neg Z_i \wedge \langle A\rangle\langle A\rangle \bigwedge_{i=0}^{2n-1} Z_i.$$

Grid coloring. To complete the reduction, we must properly define the tiling of the grid. To this end, we preliminary need to express the relations of right (horizontal) neighborhood and upper (vertical) neighborhood over the grid. We have that the following formula ψ_H (resp., ψ_V) holds over any interval $[z,t]$ such that t is the right (resp. upper) neighbor of z in $N(2^n)$:

$$\psi_H := \bigwedge_{i=n}^{2n-1} (Z_i \iff T_i) \wedge \bigwedge_{i=0}^{n-1} (T_i \iff (Z_i \oplus Z^*_i))$$

$$\psi_V := \bigwedge_{i=0}^{n-1} (Z_i \iff T_i) \wedge \bigwedge_{i=n}^{2n-1} (T_i \iff (Z_i \oplus ZH^*_i))$$

By using ψ_H and ψ_V, we can impose the *adjacency conditions* by means of the following formula (let us call ϵ):

$$[A][A]\left((\psi_H \Longrightarrow \bigvee_{(c,c')\in H} Z_c \wedge T_{c'}) \wedge (\psi_V \Longrightarrow \bigvee_{(c,c')\in V} Z_c \wedge T_{c'})\right).$$

The fact that every point is tiled by exactly one color can be forced by the following formula (let us call it ζ):

$$[A][A]\left(\dot{\bigvee_{c\in C}} Z_c \wedge \dot{\bigvee_{c\in C}} T_c\right),$$

where $\dot{\bigvee}$ is a *generalized exclusive or* which is true if and only if exactly one of its arguments is true.

Let us define φ as the conjunction $\alpha \wedge \beta \wedge \gamma \wedge \delta \wedge \epsilon \wedge \zeta$. The length of φ is polynomial in n as requested. It remains to show that φ is satisfiable if and only if $\mathcal{D}$ tiles $N(2^n)$. As for the implication from left to right, if a correct tiling exists, then let $\mathbb{D} = \langle D, <\rangle$ be a linear ordering such that:

- $D = \{d_0, d_1\} \cup N(2^n) \cup \{d_\top\}$;
- $d_0 < d_1 < (x,y) < d_\top$, for every $(x,y) \in N(2^n)$;

– given two points (x, y) and (x', y') of $N(2^n)$, $(x, y) < (x, y')$ iff $y < y' \vee (y = y' \wedge x < x')$.

Notice that we take as the domain of the interval structure the set of elements of the grid extended with the elements d_0, d_1, and $d_\top$. The elements d_0, d_1 define the initial interval $[d_0, d_1]$ over which our formula will be interpreted. The element $d_\top$ is the right endpoint of the only interval having the last point of the grid as its left endpoint, namely, $[(2^n - 1, 2^n - 1), d_\top]$.

As for the valuation $\mathcal{V}$, for any interval $[z, t]$, with $z = (x, y), t = (x', y')$, and $z, t \in N(2^n)$, $Z_i \in \mathcal{V}([z, t])$ if and only $z_i = 1$ and $T_i \in \mathcal{V}([z, t])$ if and only $t_i = 1$. Moreover, $Z_c \in \mathcal{V}([z, t])$ (resp., $T_c \in \mathcal{V}([z, t])$) if and only if $\tau(x, y) = c$ (resp., $\tau(x', y') = c$). Whenever, the left (resp. right) endpoint of an interval does not belong to $N(2^n)$, the valuation of the propositional letters Z_i and Z_c (resp. T_i and T_c) over the interval is arbitrary. It is not difficult to prove that $\mathbf{M} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V}\rangle$ is a model of φ, that is, $\mathbf{M}, [d_0, d_1] \Vdash \varphi$.

Conversely, let $\mathbf{M} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V}\rangle$ be a model for φ, that is, $\mathbf{M}, [d_0, d_1] \Vdash \varphi$. To provide a tiling of $N(2^n)$, we first define a function $f : N(2^n) \to D$ that associates a point $d \in D$ with every point $(x, y) \in N(2^n)$ in such a way that:

1. the binary representation of (x, y) coincides with the sequence of truth values of the propositional letters $Z_{2n-1}, Z_{2n-2}, \ldots, Z_0$ over the intervals $[d, d'] \in \mathbb{I}(\mathbb{D})^-$;
2. for every $(x, y), (x', y') \in N(2^n)$, $(x, y) < (x', y')$ iff $f(x, y) < f(x', y')$.

The formula φ ensures that such a function exists. Note that the definition of f guarantees commutativity: by moving first one step right and then one step up on the grid one reaches the same point that can be reached by moving first one step up and then one step right. On the basis of such a function, we define the tiling $\tau(x, y) = c$, where c is the unique element of C such that $\mathbf{M}, [f(x, y), d'] \Vdash Z_c$, for every $d' > f(x, y)$. It is not difficult to prove that τ defines a tiling of $N(2^n)$.

Putting together Theorems 32 and 33, we have the following corollary.

Corollary 7. *The satisfiability problem for* RPNL$^-$*, over natural numbers, is NEXPTIME-complete.*

6 A tableau-based decision procedure for RPNL$^-$

In this section, we define a tableau-based decision procedure for RPNL$^-$, whose behavior is illustrated by means of a simple example, and we analyze its computational complexity. Then, we prove its soundness and completeness. The procedure is based on two expansion rules, respectively called step rule and fill-in rule, and a blocking condition, that guarantees the termination of the method. Unlike the naïve procedure described in the previous section, it does not need to differentiate the search for a finite model from that for an infinite one.

6.1 The tableau method

We first define the structure of a tableau for an RPNL$^-$ formula and then we show how to construct it. A tableau for RPNL$^-$ is a suitable *decorated tree* $\mathcal{T}$. Each branch B of a tableau is associated with a finite prefix of the natural numbers $\mathbb{D}_B = \langle D_B, < \rangle$. The *decoration* of each node n in $\mathcal{T}$, denoted by $\nu(n)$, is a pair $\langle [d_i, d_j], A\rangle$, where d_i, d_j, with $d_i < d_j$, belong to D_B (for all branches B containing n) and A is an atom. The root r of $\mathcal{T}$ is labelled by the *empty decoration* $\langle \emptyset, \emptyset \rangle$. Given a node n, we denote by $A(n)$ the atom component of $\nu(n)$.

Given a branch B, we define a function $\text{REQ} : D_B \to 2^{\text{TF}(\varphi)}$ as follows. For every $d_i \in D_B$, $\text{REQ}(d_i) = (\bigcap_j A_j) \cap \text{TF}(\varphi)$, where n_j is a node such that $\nu(n_j) = \langle [d_j, d_i], A_j \rangle$ and $d_0 \leq d_j < d_i$. Moreover, given a node $n \in B$, with decoration $\langle [d_i, d_j], A \rangle$, and an existential formula $\langle A \rangle \psi \in A$, we say that $\langle A \rangle \psi$ is *fulfilled on* B if and only if there exists a node $n' \in B$ such that $\nu(n') = \langle [d_j, d_k], A' \rangle$ and $\psi \in A'$. A node n is said to be *active on* B if and only if $A(n)$ contains at least one existential formula that is not fulfilled on B.

Expansion rules. The construction of a tableau is based on the following expansion rules. Let B be a branch of a decorated tree $\mathcal{T}$ and let d_k be the greatest point in D_B. The following *expansion rules* can be possibly applied to extend B:

1. *Step rule*: if there exists at least one active node $n \in B$, with $\nu(n) = \langle [d_i, d_j], A \rangle$, then take an atom A' such that $A \; R_\varphi \; A'$ and expand B to $B \cdot n'$, with $\nu(n') = \langle [d_j, d_{k+1}], A' \rangle$.
2. *Fill-in rule*: if there exists a node $n \in B$, with decoration $\langle [d_i, d_j], A \rangle$ and $d_j < d_k$, such that there are no nodes n' in B with decoration $\langle [d_j, d_k], A' \rangle$, for some $A' \in A_\varphi$, then take any atom $A'' \in A_\varphi$ such that $A \; R_\varphi \; A''$ and $\text{REQ}(d_k) = A'' \cap \text{TF}(\varphi)$, and expand B to $B \cdot n''$, with $\nu(n'') = \langle [d_j, d_k], A'' \rangle$.

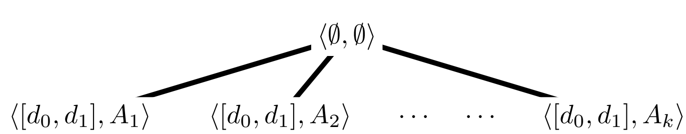
Both rules add a new node to the branch B. However, while the step rule decorates such a node with a new interval ending at a new point d_{k+1}, the fill-in rule decorates it with a new interval whose endpoints were already in D_B.

Blocking condition. To guarantee the termination of the method, we need a suitable *blocking condition* to avoid the infinite application of the expansion rules in case of infinite models. Given a branch B, with $D_B = \{d_0, d_1, \ldots, d_k\}$, we say that B is *blocked* if $\text{REQ}(d_k)$ occurs $\frac{|\text{TF}(\varphi)|}{2} + 1$ times in D_B.

Expansion strategy. Given a decorated tree $\mathcal{T}$ and a branch B, we say that an expansion rule is *applicable on* B if B is non-blocked and the application of the rule generates a new node. The *branch expansion strategy* for a branch B is the following one:

1. if the fill-in rule is applicable, apply the fill-in rule to B and, for every possible choice for the atom A'', add an immediate successor to the last node in B;
2. if the fill-in rule is not applicable and there exists a node $n \in B$, with decoration $\langle [d_i, d_j], A \rangle$ and $d_j < d_k$, such that there are no nodes in B with decoration $\langle [d_j, d_k], A' \rangle$, for some $A' \in A_\varphi$, *close* the branch;
3. if the fill-in rule is not applicable, B is not closed, and there exists at least one active node in B, then apply the step rule to B and, for every possible choice of the atom A', add an immediate successor to the last node in B.

Tableau. Let φ be the formula to be checked for satisfiability and let $A_1, \ldots, A_k$ be all and only the atoms containing φ. The *initial tableau* for φ is the following:



A *tableau* for φ is any decorated tree $\mathcal{T}$ obtained by expanding the initial tableau for φ through successive applications of the branch-expansion strategy to currently existing branches, until the branch-expansion strategy cannot be applied anymore.

Fulfilling branches. Given a branch B of a tableau $\mathcal{T}$ for φ, we say that B is a *fulfilling branch* if and only if B is not closed and one of the following conditions holds:

1. B is non-blocked and for every node $n \in B$ and existential formula $\langle A \rangle \psi \in A(n)$, there exists a node $n' \in B$ fulfilling $\langle A \rangle \psi$ (finite model case);
2. B is blocked, d_k is the greatest point of D_B, d_i ($\neq d_k$) is the smallest point in D_B such that $\mathrm{REQ}(d_i) = \mathrm{REQ}(d_k)$, and the following conditions hold:
   (a) for every node $n \in B$ and every formula $\langle A \rangle \psi \in A(n)$ not fulfilled on B, there exist a point $d_i \leq d_l \leq d_k$ and an atom A' such that $\psi \in A'$, $A(n)\ R_\varphi\ A'$, and $\mathrm{REQ}(d_l) = A' \cap \mathrm{TF}(\varphi)$;
   (b) for every node $n \in B$ and every point $d_i \leq d_m \leq d_k$, there exists an atom A' such that $A(n)\ R_\varphi\ A'$ and $\mathrm{REQ}(d_m) = A' \cap \mathrm{TF}(\varphi)$.

The decision procedure works as follows: given a formula φ, it constructs a tableau $\mathcal{T}$ for φ and it returns "satisfiable" if and only if there exists at least one fulfilling branch in $\mathcal{T}$.

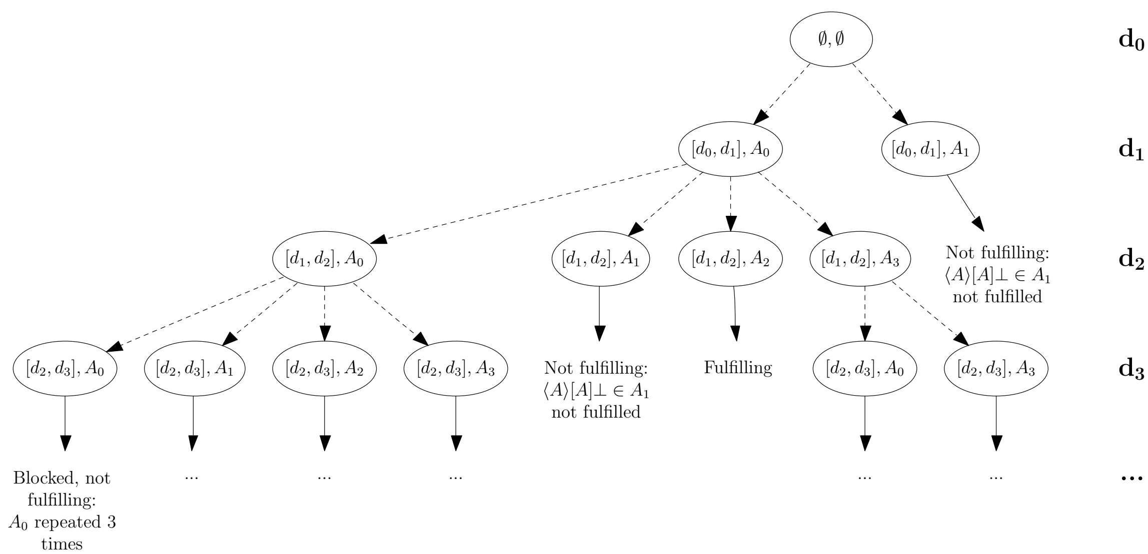


Fig. 4. A tableau for $\langle A \rangle [A] \bot$.

We conclude the section by showing how the proposed method works on the simple case of the formula $\varphi = \langle A \rangle [A] \bot$. For sake of simplicity, we treat the logical constants $\top$ and $\bot$ as propositional letters, with the further constraint that, for any atom A, $\bot \notin A$. Hence, the set of φ-atoms is the following one:

$$A_0 = \{\langle A \rangle [A] \bot, \langle A \rangle \top, \top\}$$
$$A_1 = \{\langle A \rangle [A] \bot, [A] \bot, \top\}$$
$$A_2 = \{[A] \langle A \rangle \top, [A] \bot, \top\}$$
$$A_3 = \{[A] \langle A \rangle \top, \langle A \rangle \top, \top\}$$

As for the relation R_φ between atoms, we have that

$$R_\varphi = \{(A_0, A_0), (A_0, A_1), (A_0, A_2), (A_0, A_3), (A_3, A_0), (A_3, A_3)\}.$$

Note that A_1 and A_2 have no R_φ-successors (since they both contains $[A]\bot$). Figure 4, where dashed arrows represent applications of the step rule, depicts a portion of a tableau for φ which is sufficiently large to include a fulfilling branch, and thus to prove that φ is

satisfiable. Indeed, it is easy to see that, over natural numbers, φ is satisfiable and it admits only *finite models*.
Notice that there are no nodes in the tableau which are labelled by intervals beginning at d_0, except for the nodes associated with the initial interval $[d_0, d_1]$. Since RPNL is not strong enough to force any condition on intervals beginning at d_0 and different from $[d_0, d_1]$, such intervals can be ignored without affecting the soundness and completeness of the method.

6.2 Computational complexity

As a preliminary step, we show that the proposed tableau method terminates; then we analyze its computational complexity.
In order to prove termination of the tableau method, we give a bound on the length of any branch B of any tableau for φ:

1. by the blocking condition, after at most $|REQ_\varphi| \cdot \frac{|\operatorname{TF}(\varphi)|}{2}$ applications of the step rule, the expansion strategy cannot be applied anymore to a branch;
2. given a branch B, between two successive applications of the step rule, the fill-in rule can be applied at most k times, where k is the current number of elements in D_B (k is exactly the number of applications of the step rule up to that point);
3. $|\operatorname{TF}(\varphi)|$ is linear in the length of φ, while $|REQ_\varphi|$ is exponential in it.

Hence, if $|\varphi| = n$, the length of any branch B of a tableau $\mathcal{T}$ for φ is bounded by

$$\begin{aligned} |\operatorname{REQ}_\varphi| \cdot \frac{|\operatorname{TF}(\varphi)|}{2} + \sum_{i=1}^{|\operatorname{REQ}_\varphi| \cdot \frac{|\operatorname{TF}(\varphi)|}{2}} i &\leq \left(|REQ_\varphi| \cdot \frac{|\operatorname{TF}(\varphi)|}{2}\right)^2 + 1 \\ &= \left(2^{O(n)} \cdot O(n)\right)^2 \\ &= 2^{2 \cdot O(n)} \cdot O(n^2) \\ &= 2^{O(n)} \cdot O(n^2) \\ &= 2^{O(n)}, \end{aligned}$$

that is, the length of a branch is (at most) exponential in $|\varphi|$.

Theorem 34 (Termination). *The tableau method for* RPNL^- *terminates.*

Proof. Given a formula φ, let $\mathcal{T}$ be a tableau for φ. Since, by construction, every node of $\mathcal{T}$ has a finite outgoing degree and every branch of it is of finite length, by König's Lemma, $\mathcal{T}$ is finite. □

The computational complexity of the tableau-based decision procedure depends on the strategy used to search for a fulfilling branch in the tableau. The strategy that first builds the entire tableau and then looks for a fulfilling branch requires an amount of time and space that can be doubly exponential in the length of φ. However, by exploiting nondeterminism, the existence of a fulfilling branch can be determined without visiting the entire tableau, by exploiting the following alternative strategy. First, select one of the nodes decorated with $\langle [d_0, d_1], A \rangle$ of the initial tableau and expand it as follows. Instead of generating all successors nodes, nondeterministically select one of them and expand it. Iterate such a revised expansion strategy until it cannot be applied anymore. Finally, return "satisfiable" if and only if the guessed branch is a fulfilling one.

Such a procedure has a nondeterministic time complexity which is polynomial in the length of the branch, and thus exponential in the size of φ. Giving the NEXPTIME-completeness of the satisfiability problem for RPNL^-, this allows us to conclude that the proposed tableau-based decision procedure is optimal.

6.3 Soundness and completeness

The soundness and completeness of the proposed method can be proved as follows. Soundness is proved by showing how it is possible to construct a fulfilling LIS satisfying φ from a fulfilling branch B in a tableau $\mathcal{T}$ for φ (by Theorem 29, it follows that φ has a model). The proof must encompass both the case of blocked branches and that of non-blocked ones. Proving completeness consists in showing, by induction on the height of $\mathcal{T}$, that for any satisfiable formula φ, there exists a fulfilling branch B in any tableau $\mathcal{T}$ for φ. To this end, we take a model for φ and the corresponding fulfilling LIS $\mathbf{L}$, and we prove the existence of a fulfilling branch in $\mathcal{T}$ by exploiting Theorems 30 and 31.

Theorem 35 (Soundness). *Given a formula φ and a tableau $\mathcal{T}$ for φ, if there exists a fulfilling branch in $\mathcal{T}$, then φ is satisfiable.*

Proof. Let $\mathcal{T}$ be a tableau for φ and B a fulfilling branch in $\mathcal{T}$. We show that, starting from B, we can build up a fulfilling LIS $\mathbf{L}$ satisfying φ. By the definition of fulfilling branch, two cases may arise.

B is non-blocked (*finite model case*). We define $\mathbf{L} = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ as follows:
- $\mathbb{D} = \mathbb{D}_B$;
- $\mathcal{L}([d_0, d_1]) = A$, where $A = A(n_1)$ and n_1 is the unique node of B such that $\nu(n_1) = \langle[d_0, d_1], A\rangle$;
- for every $[d_i, d_j] \in \mathbb{I}(\mathbb{D}_B)^-$, with $d_i > d_0$, we put $\mathcal{L}([d_i, d_j]) = A$, where $A = A(n)$ and n is the node in B such that $\nu(n) = \langle[d_i, d_j], A\rangle$. Since B is not closed, such a node n exists; its uniqueness follows from tableau rules;
- finally, for every interval $[d_0, d_j] \in \mathbb{I}(\mathbb{D}_B)^-$, with $d_j > d_1$, there are no nodes in B labelled with $\langle[d_0, d_j], A\rangle$. We complete the definition of $\mathcal{L}$ by putting, for every $d_j > d_1$, $\mathcal{L}([d_0, d_j]) = A(n)$, where n is an arbitrary node in B such that $\nu(n) = \langle[d_i, d_j], A(n)\rangle$, for some $d_i < d_j$.

Clearly, $\mathbf{L}$ is a LIS. Since B is fulfilling, for every node $n \in B$ and every existential formula $\langle A\rangle\psi \in A(n)$, there exists a node n' fulfilling $\langle A\rangle\psi$. Hence, by the above construction, $\mathbf{L}$ is fulfilling.

B is blocked (*infinite model case*). Let d_k be the last point of D_B and $d_i \neq d_k$ be the smallest point in D_B such that $\text{REQ}(d_i) = \text{REQ}(d_k)$. We build an ultimately periodic LIS $\mathbf{L} = \langle\langle\mathbb{D}', \mathbb{I}(\mathbb{D}')^-\rangle, \mathcal{L}\rangle$ with prefix $l = i - 1$ and period $p = k - i$, as follows:

1. let $D' = \{d'_0(= d_0), d'_1(= d_1), \ldots, d'_k(= d_k), d'_{k+1}, \ldots\}$ be any set isomorphic to $\mathbb{N}$;
2. for every $d'_1 \leq d'_j \leq d'_k$, put $\text{REQ}(d'_j) = \text{REQ}(d_j)$;
3. for every $d'_j > d'_k$, put $\text{REQ}(d'_j) = \text{REQ}(d_{i+(j-l) \,\texttt{mod}\, p})$;
4. put $\mathcal{L}([d'_0, d'_1]) = A(n_1)$, where n_1 is the unique node in B such that $\nu(n_1) = \langle[d_0, d_1], A\rangle$;
5. for every pair of points d'_j, d'_m such that $d_0 < d'_j < d'_m < d'_k$, take the node n in B such that $\nu(n) = \langle[d_j, d_m], A\rangle$ and put $\mathcal{L}([d'_j, d'_m]) = A$;

6. for every point $d'_j \in D'$ and every $\langle A\rangle\psi \in \mathrm{REQ}(d'_j)$ which has not been fulfilled yet, proceed as follows. Let n be a node in B decorated with $\langle [d, d'], A\rangle$ such that $\mathrm{REQ}(d') = \mathrm{REQ}(d'_j)$. Since B is fulfilling, by condition (a) for fulfilling branches, there exist a point $d_i \leq d_m \leq d_k$ and an atom A' such that $\psi \in A'$, $A\ R_\varphi\ A'$ and $\mathrm{REQ}(d_m) = A' \cap \mathrm{TF}(\varphi)$. By the definition of $\mathbf{L}$, we have that there exist infinitely many points $d'_n \geq d'_k$ in D' such that $\mathrm{REQ}(d'_n) = \mathrm{REQ}(d_m)$. We can take one of such points d'_n such that $\mathcal{L}([d'_j, d'_n])$ has not been defined yet and put $\mathcal{L}([d'_j, d'_n]) = A'$;
7. once we have fulfilled all $\langle A\rangle$-formulae in $\mathrm{REQ}(d')$, for all $d' \in \mathbb{D}'$, we arbitrarily define the labelling of the remaining intervals $[d', d'']$. Since B is fulfilling, we can always define $\mathcal{L}([d', d''])$ by exploiting condition (b) for fulfilling branches;
8. as in the finite model case, there are no nodes in B labelled with $\langle [d_0, d_j], A\rangle$, for all $d_j > d_1$. For every $d'_j > d'_1$, take an arbitrary node n labelled with $\langle [d_i, d_l], A'\rangle$ such that $\mathrm{REQ}(d_l) = \mathrm{REQ}(d'_j)$ and put $\mathcal{L}([d'_0, d'_j]) = A'$.

Since $\varphi \in \mathcal{L}([d'_0, d'_1])$, $\mathbf{L}$ is a fulfilling LIS satisfying φ. □

Theorem 36 (Completeness). *Given a satisfiable formula φ, there exists a fulfilling branch in every tableau $\mathcal{T}$ for φ.*

Proof. Let φ be a satisfiable formula and let $\mathbf{L} = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{L}\rangle$ be a fulfilling LIS satisfying φ, whose existence is guaranteed by Theorem 29. Without loss of generality, we may assume that $\mathbf{L}$ respects the constraints of Theorem 30, if it is finite, and of Theorem 31, if it is infinite. We prove there exists a fulfilling branch B in $\mathcal{T}$ which corresponds to $\mathbf{L}$. To this end, we prove the following property: *there exists a non-closed branch B such that, for every node $n \in B$, if n is decorated with $\langle [d_j, d_k], A\rangle$, then $A = \mathcal{L}([d_j, d_k])$*. The proof is by induction on the height $h(\mathcal{T})$ of $\mathcal{T}$.

If $h(\mathcal{T}) = 1$, then $\mathcal{T}$ is the initial tableau for φ and, by construction, it contains a branch

$$B_0 = \langle \emptyset, \emptyset\rangle \cdot \langle [d_0, d_1], A\rangle,$$

with $A = \mathcal{L}([d_0, d_1])$.
Let $h(\mathcal{T}) = i + 1$. By inductive hypothesis, there exists a branch B_i of length i that satisfies the property. Let $D_{B_i} = \{d_0, d_1, \ldots, d_k\}$. We distinguish two cases, depending on the expansion rule that has been applied to B_i in the construction of $\mathcal{T}$.

- **The step rule has been applied.**
  Let n be the active node, decorated with $\langle [d_j, d_l], A\rangle$, which the step rule has been applied to. By inductive hypothesis, $A = \mathcal{L}([d_j, d_l])$. Since $\mathbf{L}$ is a LIS, $\mathcal{L}([d_j, d_l])\ R_\varphi\ \mathcal{L}([d_l, d_{k+1}])$. Hence, there must exists in $\mathcal{T}$ a successor n' of the last node of B_i decorated with $\langle [d_l, d_{k+1}], \mathcal{L}([d_l, d_{k+1}])\rangle$. Let $B_{i+1} = B_i \cdot \langle [d_l, d_{k+1}], \mathcal{L}([d_l, d_{k+1}])\rangle$. Since the step rule can be been applied only to non-closed branches (and it does not close any branch), B_{i+1} is non-closed.
- **The fill-in rule has been applied.**
  Let n be the node decorated with $\langle [d_j, d_l], A\rangle$ such that there exist no nodes in B_i decorated with $\langle [d_l, d_k], A'\rangle$ for some atom A'. By inductive hypothesis, $A = \mathcal{L}([d_j, d_l])$. Since $\mathbf{L}$ is a LIS, $\mathcal{L}([d_j, d_l])\ R_\varphi\ \mathcal{L}([d_l, d_k])$. Hence, there must exists in $\mathcal{T}$ a successor of the last node of B_i decorated with $\langle [d_l, d_k], \mathcal{L}([d_l, d_k])\rangle$. Let $B_{i+1} = B_i \cdot \langle [d_l, d_k], \mathcal{L}([d_l, d_k])\rangle$. As before, since the fill-in rule can be applied only to non-closed branches (and it does not close any branch), B_{i+1} is not closed.

Now we show that B is the fulfilling branch we are searching for. Since B is not closed, two cases may arise.

- *B is non-blocked and the expansion strategy cannot be applied anymore.* Since B is not closed, this means that there exist no active nodes in B, that is, for every node $n \in B$ and every formula $\langle A \rangle \psi \in A(n)$, there exists a node n' fulfilling $\langle A \rangle \psi$. Hence, B is a fulfilling branch.
- *B is blocked.* This implies that $\text{REQ}(d_k)$ is repeated $\frac{|\text{TF}(\varphi)|}{2} + 1$ times in B. Since B is decorated coherently to $\mathbf{L}$ from d_0 to d_k, by Theorem 30, we can assume $\mathbf{L}$ to be infinite. Let d_j be the smallest point in D_B such that $\text{REQ}(d_j) = \text{REQ}(d_k)$. We have that $\mathbf{L}$ is ultimately periodic, with prefix $l = j - 1$, since (by Theorem 31) the only set of requests which has been repeated $\frac{|\text{TF}(\varphi)|}{2} + 1$ times in B is the one associated with the first point in the period. Furthermore, we have that, between d_{l+1} and d_{k-1}, there are exactly $\frac{|\text{TF}(\varphi)|}{2}$ repetitions of the period of $\mathbf{L}$. This allows us to exploit the structural properties of $\mathbf{L}$ to prove that B is fulfilling.
  For every node $n \in B$ decorated with $\langle [d, d'], A \rangle$ and for every formula $\langle A \rangle \psi \in A$, since $\mathbf{L}$ is fulfilling, there exists a point d'' in D such that $\psi \in \mathcal{L}([d', d''])$. If $d'' \leq d_k$, then $\langle A \rangle \psi$ is fulfilled in B. Otherwise, there exists some point d_m, with $d_i \leq d_m \leq d_k$, such that $\text{REQ}(d'') = \text{REQ}(d_m)$. Hence, the atom $A' = \mathcal{L}([d', d''])$ can be chosen in order to satisfy condition (a) of the definition of fulfilling branch.
  For every node $n \in B$ decorated with $\langle [d, d'], A \rangle$ and for every point $d_j \leq d_m \leq d_k$, we have that $\text{REQ}(d_m) \in \mathit{Inf}(\mathbf{L})$. Hence, there exist infinitely many points d_n in $\mathbf{L}$ such that $\text{REQ}(d_m) = \text{REQ}(d_n)$ and $d' < d_n$. Let d_n be one of such points. We can choose the atom $A' = \mathcal{L}([d', d_n])$ to satisfy condition (b) of the definition of fulfilling branch. □

7 A tableau-based decision procedure for RPNL$^+$

In this section we briefly show how to adapt the decision procedure for RPNL$^-$ to RPNL$^+$. First of all, we define the notion of *non-strict* φ-labelled interval structure as follows.

Definition 27. *A* non-strict φ-labelled interval structure *(non-strict* LIS*, for short) is a pair* $\mathbf{L} = \langle \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle, \mathcal{L} \rangle$*, where* $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$ *is a non-strict interval structure and* $\mathcal{L} : \mathbb{I}(\mathbb{D})^+ \to A_\varphi$ *is a* labelling function *such that, for every pair of neighboring intervals* $[d_i, d_j], [d_j, d_k] \in \mathbb{I}(\mathbb{D})^+$*,* $\mathcal{L}([d_i, d_j]) \; R_\varphi \; \mathcal{L}([d_j, d_k])$*.*

It is possible to prove that Theorems 29, 30, and 31 hold also for non-strict LISs. Furthermore, we can easily tailor the tableau-based decision method for RPNL$^-$ to RPNL$^+$ by adding the following expansion rule:

3. *Point-intervals rule*: if d_k is the last point of D_B and there exists a node $n \in B$ decorated with $\langle [d_j, d_k], A \rangle$, with $d_j < d_k$, such that there are no nodes in B decorated with $\langle [d_k, d_k], A' \rangle$ for some $A' \in A_\varphi$, then take any atom $A'' \in A_\varphi$ such that $A \; R_\varphi \; A''$ and $\text{REQ}(d_k) = A'' \cap \text{TF}(\varphi)$, and expand B to $B \cdot n'$, with $\nu(n') = \langle [d_k, d_k], A'' \rangle$.

The expansion strategy has to be expanded accordingly to take into account this new rule; on the contrary, the blocking condition, the definition of initial tableau, and the definition of fulfilling branch remain unchanged. Termination, soundness, and completeness of the resulting tableau method for RPNL$^+$ can be proved as in the case of RPNL$^-$.
Finally, to prove the optimality of the tableau for RPNL$^+$, we can exploit the reduction given in Section 5, provided that we replace $\langle A \rangle$ by $\Diamond_r$ and $[A]$ by $\Box_r$.

Theorem 37. *The satisfiability problem for* RPNL$^+$*, over natural numbers, is NEXPTIME-complete.*

8 Conclusions

In this paper, we focussed our attention on interval logics of temporal neighborhood. We addressed the satisfiability problem for the future fragment of strict Neighborhood Logic (RPNL$^-$), interpreted over natural numbers, and we showed that it is NEXPTIME-complete. In particular, we proved NEXPTIME-hardness by a reduction from the exponential tiling problem. Then, we developed a sound and complete tableau-based decision procedure for RPNL$^-$ and we proved its optimality. We concluded the paper by briefly showing that such a procedure can be easily adapted to non-strict RPNL (RPNL$^+$).
The proposed decision procedure improves the EXPSPACE tableau-based decision method for checking RPNL$^-$ satisfiability developed by Bresolin and Montanari in [18]. We do not see any relevant problem in adapting our procedure to deal with the satisfiability problem for RPNL$^-$ interpreted over branching temporal structures (where every branch is isomorphic to natural numbers) [19]. Furthermore, we believe it possible to generalize it to cope with Branching Time Neighborhood Logic [17], a branching-time interval neighborhood logic that interleaves operators that quantify over possible branches with operators that quantify over intervals belonging to a given branch. On the contrary, the extension to full PNL turned out to be much more difficult. In particular, there is not a straightforward way of generalizing the basic removal technique exploited by Theorems 30 and 31 to bound the search space. In the presence of past operators, indeed, the removal of a point may affect both future existential formulae and past existential ones, and there is not an easy way to fix the future and past *defects* (see Section 4) it may introduce.

Copy of: **D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Complete and terminating tableau for the logic of proper subinterval structures over dense orderings. Proceedings of M4M 4: 5th Workshop on Methods for Modalities, C. Areces and S. Demri (Eds.), November 2007, pp. 335-351 (to appear in the Electronic Notes in Theoretical Computer Science)**

Abstract. We introduce special pseudo-models for the interval logic of proper subintervals over dense linear orderings. We prove finite model property with respect to such pseudo-models, and using that result we develop a decision procedure based on a sound, complete, and terminating tableau for that logic. The case of proper subintervals is essentially more complicated than the case of strict subintervals, for which we developed a similar tableau-based decision procedure in a recent work.

1 Introduction

In interval temporal logics undecidability is usually the case (see, for instance, [59,74]), while decidability is a rare exception. The quest for decidable fragments and systems of temporal logics with interval-based semantics is one of the main research problems in the area of interval logics. Several decidability results have been established previously by reduction to point-based logics, either by way of direct translation or by restriction of the semantics, e.g., imposing locality, homogeneity, or other principles that essentially reduce it to point-based semantics [1,10,11,115,33,68,82]
Only recently some decidability results of genuinely interval-based logics have been established [13,15,17,18,20,21]. In particular, in [13] we have developed a sound, complete and terminating tableau for the logic $\mathrm{D}_{\dot\sqsubset}$ of *strict subintervals* (with both endpoints strictly inside the current interval) over dense linear orderings, by defining a class of pseudo-models and proving finite model property with respect to such pseudo-models.
Here we consider the interval logic $\mathrm{D}_{\sqsubset}$ of *proper subintervals*, that is, subintervals different from the current interval, over dense linear orderings and we develop a similar technique to devise a tableau-based decision procedure for that logic. Despite the strong similarity with our previous work, the case of proper subintervals turned out to be essentially more complicated. The presence of the special families of beginning subintervals and ending subintervals of a given interval in a structure with proper subinterval relation causes substantial distinction of the semantics from the case of interval structures with strict subinterval relation studied in [13], further leading to considerable complications in the constructions of both pseudo-models and tableaux. For instance, the formula $(\langle D\rangle p \wedge \langle D\rangle q) \rightarrow \langle D\rangle(\langle D\rangle p \wedge \langle D\rangle q)$ is valid in $\mathrm{D}_{\dot\sqsubset}$ but not in $\mathrm{D}_{\sqsubset}$ (for, p and q may only be satisfied in respectively beginning and ending subintervals). Furthermore, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

can only be satisfied in a $\mathrm{D}_{\sqsubset}$-structure, as it forces p to be true at some beginning and at some ending subintervals, a requirement which cannot be imposed in $\mathrm{D}_{\dot\sqsubset}$. Note, however, that while $\mathrm{D}_{\sqsubset}$ can refer to beginning or ending intervals, *it cannot differentiate between these*. This is a subtle but crucial detail: as shown by Lodaya [74], the interval logic BE with modalities respectively for beginning and ending subintervals is *undecidable over the class of dense orderings*.

The paper is organized as follows. In Section 2, we give the syntax and semantics of the logic of proper subintervals $\mathrm{D}_{\sqsubset}$. Moreover, we introduce pseudo-models for $\mathrm{D}_{\sqsubset}$ and we prove that satisfiability of $D_\sqsubset$-formulas in pseudo-models is equivalent to satisfiability in standard models, thus establishing a small model property for $D_\sqsubset$. Section 3 is devoted to the tableau-based decision procedure obtained from the latter result. We conclude the paper with a short discussion of related open problems and future research.

2 Structures for $D_\sqsubset$ formulas

2.1 Syntax and semantics of $D_\sqsubset$

Let $\mathbb{D} = \langle D, < \rangle$ be a dense linear order. An *interval* over $\mathbb{D}$ is an ordered pair $[b, e]$, where $b < e$. We denote the set of all intervals over $\mathbb{D}$ by $\mathbb{I}(\mathbb{D})$. We consider the *proper* (i.e., irreflexive) *subinterval relation*, denoted by $\sqsubset$, defined as follows: $[d_k, d_l] \sqsubset [d_i, d_j]$ if and only if $d_i \leq d_k$, $d_l \leq d_j$ and $[d_k, d_l] \neq [d_i, d_j]$. We shall write $[d_k, d_l] \sqsubseteq [d_i, d_j]$ as a shorthand for $[d_k, d_l] \sqsubset [d_i, d_j] \vee [d_k, d_l] = [d_i, d_j]$.
The language of the modal logic $D_\sqsubset$ of interval structures with proper subinterval relation consists of a set $\mathcal{AP}$ of propositional letters, the propositional connectives $\neg$ and $\vee$, and the modal operator $\langle D \rangle$. The other propositional connectives, as well as the logical constants $\top$ (*true*) and $\bot$ (*false*) and the dual modal operator $[D]$, are defined as usual. Formulas of $D_\sqsubset$ are defined as follows: $\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle D \rangle \varphi$. The semantics of $D_\sqsubset$ is based on *interval models* $\mathbf{M} = \langle \mathbb{I}(\mathbb{D}), \sqsubset, \mathcal{V} \rangle$. The *valuation function* $\mathcal{V} : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})}$ assigns to every propositional variable p the set of intervals $\mathcal{V}(p)$ over which p holds. The semantics of $D_\sqsubset$ is recursively defined by the satisfiability relation $\Vdash$ as follows:

- for every propositional variable $p \in \mathcal{AP}$, $\mathbf{M}, [d_i, d_j] \Vdash p$ iff $[d_i, d_j] \in \mathcal{V}(p)$;
- $\mathbf{M}, [d_i, d_j] \Vdash \neg\psi$ iff $\mathbf{M}, [d_i, d_j] \nVdash \psi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \psi_1 \vee \psi_2$ iff $\mathbf{M}, [d_i, d_j] \Vdash \psi_1$ or $\mathbf{M}, [d_i, d_j] \Vdash \psi_2$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle D \rangle \psi$ iff there exists $[d_k, d_l] \in \mathbb{I}(\mathbb{D})$ such that $[d_k, d_l] \sqsubset [d_i, d_j]$ and $\mathbf{M}, [d_k, d_l] \Vdash \psi$.

A $D_\sqsubset$-formula is *satisfiable* if it is true at some interval in some interval model; it is *valid* if it is true at every interval in every interval model.

2.2 Fulfilling $D_\sqsubset$-structures

In this section we introduce suitable pseudo-models, called *fulfilling* $D_\sqsubset$*-structures*, for $D_\sqsubset$-formulas.

Definition 28. *Given a* $D_\sqsubset$*-formula* φ*, a* φ-atom *is a subset* A *of* $\mathrm{CL}(\varphi)$ *such that:*
(i) for every $\psi \in \mathrm{CL}(\varphi)$*,* $\psi \in A$ *if and only if* $\neg\psi \notin A$*, and*
(ii) for every $\psi_1 \vee \psi_2 \in \mathrm{CL}(\varphi)$*,* $\psi_1 \vee \psi_2 \in A$ *if and only if* $\psi_1 \in A$ *or* $\psi_2 \in A$*.*

Definition 29. *Given a* $D_\sqsubset$*-formula* φ *and a* φ*-atom* $A \in \mathcal{A}_\varphi$*, the* set $\mathrm{REQ}(A)$ of (temporal) requests of A *is the set* $\{\langle D \rangle \psi \in \mathrm{CL}(\varphi) : \langle D \rangle \psi \in A\}$*.*

We denote the set of all φ-atoms by A_φ and the set of all $\langle D \rangle$-formulas in $CL(\varphi)$ by REQ_φ. Then, we define the binary relation $D_\varphi \subseteq A_\varphi \times A_\varphi$, such that $A \; D_\varphi \; A'$ if and only if for every $[D]\psi$ in $CL(\varphi)$, if $[D]\psi \in A$, then $\psi \in A'$.
Given an interval $[b, e]$, a *beginning subinterval* of $[b, e]$ is an interval $[b, e']$, with $e' < e$, an *ending subinterval* of $[b, e]$ is an interval $[b', e]$, with $b < b'$, and an *internal subinterval* of $[b, e]$ is an interval $[b', e']$, with $b < b'$ and $e' < e$. To represent infinite chains of beginning

(resp., ending) subintervals of a given interval, we need to introduce the notion of *cluster* of reflexive nodes. Given a graph $\mathbb{G} = \langle V, E \rangle$, we define a *cluster* as a maximal strongly connected subgraph $\mathcal{C}$ which includes reflexive vertices only. By abuse of notation, we say that a *cluster* $\mathcal{C}$ *is a successor of a vertex* v if v does not belong to $\mathcal{C}$ and there exists a successor v' of v in $\mathcal{C}$. Conversely, a *vertex* v *is a successor of* $\mathcal{C}$ if v does not belong to $\mathcal{C}$ and there exists a predecessor v' of v in $\mathcal{C}$. $\mathrm{D}_\sqsubset$-graphs are defined as follows.

Definition 30. *A finite directed graph* $\mathbb{G} = \langle V, E \rangle$ *is a* $\mathrm{D}_\sqsubset$-graph *if:*
1. *there exists an irreflexive vertex* $v_0 \in V$*, called the* root *of* $\mathbb{G}$*, such that any other vertex* $v \in V$ *is reachable from it;*
2. *every irreflexive vertex* $v \in V$ *has exactly two clusters as successors: a* beginning successor cluster $\mathcal{C}_b$ *and an* ending successor cluster $\mathcal{C}_e$*;*
3. $\mathcal{C}_b$ *and* $\mathcal{C}_e$ *have a unique common successor* v_c*, which is a reflexive vertex;*
4. *every successor of* v_c*, different from* v_c *itself, is irreflexive;*
5. *there exists at most one edge exiting the clusters* $\mathcal{C}_b$ *and* $\mathcal{C}_e$ *and reaching an irreflexive node;*
6. *apart from the edge leading to* v_c*, there are no edges exiting from* $\mathcal{C}_b$ *(resp.* $\mathcal{C}_e$*) that reach a reflexive vertex.*

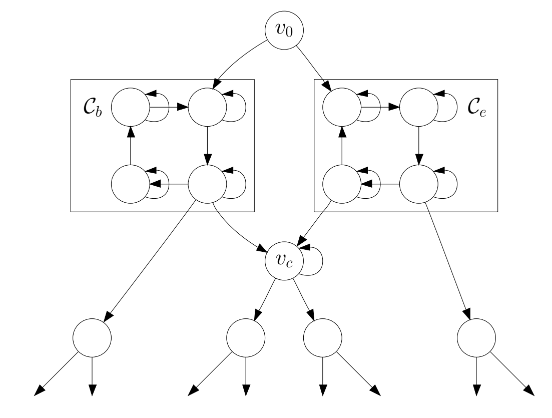


Fig. 5. An example of $\mathrm{D}_\sqsubset$-graph.

Figure 5 depicts a portion of a $\mathrm{D}_\sqsubset$-graph. The root v_0 has two successor clusters $\mathcal{C}_b$ and $\mathcal{C}_e$ of four vertices each. Both $\mathcal{C}_b$ and $\mathcal{C}_e$ have exactly one irreflexive successor. Their common reflexive successor v_c has two irreflexive successors.

Let φ be a $\mathrm{D}_\sqsubset$ formula. $\mathrm{D}_\sqsubset$-structures are defined by pairing a $\mathrm{D}_\sqsubset$-graph with a labeling function that associates an $\mathcal{A}_\varphi$ atom with each vertex of the graph.

Definition 31. *A* $\mathrm{D}_\sqsubset$-structure *is a quadruple* $\mathbf{S} = \langle \langle V, E \rangle, \mathcal{L}, \mathcal{B}, \mathcal{E} \rangle$*, where:*
1. $\langle V, E \rangle$ *is a* $\mathrm{D}_\sqsubset$*-graph;*
2. $\mathcal{L} : V \to \mathcal{A}_\varphi$ *is a* labeling function *that assigns to every vertex* $v \in V$ *an atom* $\mathcal{L}(v)$ *such that for every edge* $(v, v') \in E$*,* $\mathcal{L}(v)\ D_\varphi\ \mathcal{L}(v')$*;*
3. $\mathcal{B} : V \to 2^{\mathrm{REQ}_\varphi}$ *and* $\mathcal{E} : V \to 2^{\mathrm{REQ}_\varphi}$ *are mappings that assign to every vertex the sets of its* beginning *and* ending requests*, respectively;*
4. *for every irreflexive vertex* $v \in V$*, with successor clusters* $\mathcal{C}_b$ *and* $\mathcal{C}_e$*, we have that:*

– *the common reflexive successor* v_c *of* $\mathcal{C}_b$ *and* $\mathcal{C}_e$ *is such that* $\mathcal{E}(v_c) = \mathcal{B}(v_c) = \emptyset$ *and* $REQ(\mathcal{L}(v_c)) = REQ(\mathcal{L}(v)) - (\mathcal{B}(v) \cup \mathcal{E}(v))$,
– *every reflexive vertex* $v' \in \mathcal{C}_b$ *is such that* $\mathcal{B}(v') = \mathcal{B}(v)$, $\mathcal{E}(v') = \emptyset$, *and* $REQ(\mathcal{L}(v')) = REQ(\mathcal{L}(v_c)) \cup \mathcal{B}(v)$,
– *the unique irreflexive successor* v'' *of* $\mathcal{C}_b$ *(if any) is such that* $\mathcal{B}(v) \cap \mathcal{L}(v'') \subseteq \mathcal{B}(v'')$ *(requests which have been classified as initial in a given vertex cannot be reclassified in its descendants),*
– *every reflexive vertex* $v' \in \mathcal{C}_e$ *is such that* $\mathcal{E}(v') = \mathcal{E}(v)$, $\mathcal{B}(v') = \emptyset$, *and* $REQ(\mathcal{L}(v')) = REQ(\mathcal{L}(v_c)) \cup \mathcal{E}(v)$,
– *the unique irreflexive successor* v'' *of* $\mathcal{C}_e$ *(if any) is such that* $\mathcal{E}(v) \cap \mathcal{L}(v'') \subseteq \mathcal{E}(v'')$ *(requests which have been classified as ending in a given vertex cannot be reclassified in its descendants).*

Let v_0 *be the root of* $\langle V, E\rangle$. *If* $\varphi \in \mathcal{L}(v_0)$, *we say that* $\mathbf{S}$ *is a* $\mathrm{D}_\sqsubset$-structure for φ.

Beginning and ending requests associated with a vertex v can be viewed as requests that must be satisfied over respectively beginning and ending subintervals of any interval corresponding to v (possibly over both of them), but not over its internal subintervals.
Every $\mathrm{D}_\sqsubset$-structure can be regarded as a Kripke model for $\mathrm{D}_\sqsubset$, where the valuation is determined by the labeling.

Definition 32. *A* $\mathrm{D}_\sqsubset$*-structure* $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ *is* fulfilling *if for every* $v \in V$ *and every* $\langle D\rangle\psi \in \mathcal{L}(v)$, *there exists* $v' \in V$ *such that* v' *is a descendant of* v *and* $\psi \in \mathcal{L}(v')$.

Theorem 38. *Let* φ *be a* $\mathrm{D}_\sqsubset$*-formula which is satisfied in an interval model. Then, there exists a fulfilling* $\mathrm{D}_\sqsubset$*-structure* $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ *for* φ.

Proof. Let $\mathbf{M} = \langle \mathbb{I}(\mathbb{D}), \sqsubset, \mathcal{V}\rangle$ be an interval model and let $[b_0, e_0] \in \mathbb{I}(\mathbb{D})$ be an interval such that $\mathbf{M}, [b_0, e_0] \Vdash \varphi$. We recursively build a fulfilling $\mathrm{D}_\sqsubset$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ for φ as follows.
We start with the one-node graph $\langle\{v_0\}, \emptyset\rangle$ and a labeling function $\mathcal{L}$ such that $\mathcal{L}(v_0) = \{\psi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_0, e_0] \Vdash \psi\}$. Then, we partition the set $\mathrm{REQ}(\mathcal{L}(v_0))$ into the following three sets of formulas:

Beginning requests: B_{v_0} contains all $\langle D\rangle\xi \in \mathrm{REQ}(\mathcal{L}(v_0))$ such that ξ is satisfied over beginning subintervals of $[b_0, e_0]$, but not over internal subintervals of $[b_0, e_0]$;
Ending requests: E_{v_0} contains all $\langle D\rangle\xi \in \mathrm{REQ}(\mathcal{L}(v_0))$ such that ξ is satisfied over ending subintervals of $[b_0, e_0]$, but not over internal subintervals of $[b_0, e_0]$;
Internal requests: $I_{v_0} = (\mathrm{REQ}(\mathcal{L}(v_0)) \setminus B_{v_0}) \setminus E_{v_0}$, that is, the set of all $\langle D\rangle\xi \in \mathrm{REQ}(\mathcal{L}(v_0))$ such that ξ is satisfied over internal subintervals of $[b_0, e_0]$.

We put $\mathcal{B}(v_0) = B_{v_0}$ and $\mathcal{E}(v_0) = E_{v_0}$. Then, for every formula $\langle D\rangle\psi \in \mathcal{L}(v_0)$, we choose an interval $[b_\psi, e_\psi]$, with $[b_\psi, e_\psi] \sqsubset [b_0, e_0]$, such that $\mathbf{M}, [b_\psi, e_\psi] \Vdash \psi$. If $\langle D\rangle\psi \in I_{v_0}$, then $b_0 < b_\psi < e_\psi < e_0$, else if $\langle D\rangle\psi \in B_{v_0}$, then $b_0 = b_\psi < e_\psi < e_0$, otherwise ($\langle D\rangle\psi \in E_{v_0}$) $b_0 < b_\psi < e_\psi = e_0$.
Since $\mathbb{D}$ is a dense ordering and $\mathrm{CL}(\varphi)$ is a finite set of formulas, there exist two beginning intervals $[b_0, e_1]$ and $[b_0, e_2]$ such that:

– for every interval $[b_\psi, e_\psi]$, with $\langle D\rangle\psi \in B_{v_0} \cup I_{v_0}$, $[b_\psi, e_\psi] \sqsubset [b_0, e_2] \sqsubset [b_0, e_1]$;
– $[b_0, e_1]$ and $[b_0, e_2]$ satisfy the same formulas of $\mathrm{CL}(\varphi)$.

We start the construction of the beginning successor cluster $\mathcal{C}_b$ of v_0 by adding a new vertex v_b and a pair of edges (v_0, v_b) and (v_b, v_b), and by putting $\mathcal{L}(v_b) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_0, e_1] \Vdash \xi\}$, $\mathcal{B}(v_b) = B_{v_0}$ and $\mathcal{E}(v_b) = \emptyset$. Next, for every $\langle D\rangle\psi \in \mathcal{B}(v_b)$, we establish whether or not we must add a vertex v_ψ in $\mathcal{C}_b$ as follows. Let $[b_0, e_\psi]$ be a beginning subinterval such that $\mathbf{M}, [b_0, e_\psi] \Vdash \psi$. We add a reflexive vertex v_ψ to $\mathcal{C}_b$ if $[b_0, e_\psi]$ satisfies the same temporal formulas $[b_0, e_1]$ satisfies. Moreover, we put $\mathcal{L}(v_\psi) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_0, e_\psi] \Vdash \xi\}$, $\mathcal{B}(v_\psi) = \mathcal{B}(v_b)$, and $\mathcal{E}(v_\psi) = \emptyset$. Let $\{v_1, \ldots, v_k\}$ be the resulting set of vertices added to $\mathcal{C}_b$. For $i = 1, \ldots, k-1$, we add an edge (v_i, v_{i+1}) to E; furthermore, we add the edges (v_b, v_1) and (v_k, v_b) to E. If for all formulas $\langle D\rangle\psi \in \mathcal{B}(v_b)$ there exists a corresponding vertex v_ψ in $\mathcal{C}_b$, we are done. Otherwise, let Γ_B be the set of the remaining formulas $\langle D\rangle\psi \in \mathcal{B}(v_b)$ and let $[b_0, e_B^{max}]$ be a beginning subinterval such that, for every formula $\langle D\rangle\psi \in \Gamma_B$, we have that $\mathbf{M}, [b_0, e_B^{max}] \Vdash \psi$ or $\mathbf{M}, [b_0, e_B^{max}] \Vdash \langle D\rangle\psi$. We add a new irreflexive vertex v_b^{max} and an edge connecting an arbitrary vertex in $\mathcal{C}_b$ to it, say (v_b, v_b^{max}), and we define its labeling as $\mathcal{L}(v_b^{max}) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_0, e_B^{max}] \Vdash \xi\}$.
The ending successor cluster $\mathcal{C}_e$ of v_0 is built in the very same way.
To complete the first phase of the construction, we must introduce the common reflexive successor v_c of $\mathcal{C}_b$ and $\mathcal{C}_e$. Since $\mathbb{D}$ is a dense ordering and $\mathrm{CL}(\varphi)$ is a finite set of formulas, there exist two intervals $[b_3, e_3]$ and $[b_4, e_4]$ such that:

- for every interval $[b_\psi, e_\psi]$, with $\langle D\rangle\psi \in I_{v_0}$, $[b_\psi, e_\psi] \sqsubset [b_4, e_4] \sqsubset [b_3, e_3]$;
- $[b_3, e_3]$ and $[b_4, e_4]$ satisfy the same formulas of $\mathrm{CL}(\varphi)$.

We add a new vertex v_c, together with the edges (v_b, v_c), (v_e, v_c), and (v_c, v_c), and we put $\mathcal{L}(v_c) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_3, e_3] \Vdash \xi\}$, $\mathcal{B}(v_c) = \mathcal{E}(v_c) = \emptyset$.
For every formula $\langle D\rangle\psi \in I_{v_0}$, we add a new vertex v_ψ and an edge (v_c, v_ψ), and we define its labeling as $\mathcal{L}(v_\psi) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_\psi, e_\psi] \Vdash \xi\}$.
Then, we recursively apply the above procedure to the irreflexive vertices we have introduced. To keep the construction finite, whenever there exists an irreflexive vertex $v' \in V$ such that $\mathcal{L}(v_\psi) = \mathcal{L}(v')$ for some v_ψ, we simply add an edge to v' instead of creating a new vertex v_ψ and an edge entering it. Since the set of atoms is finite, the construction is guaranteed to terminate.

Let $\mathbf{S}$ be a fulfilling $\mathrm{D}_\sqsubset$-structure for a formula φ. To build a model for φ, we consider the interval $[0, 1]$ of the rational line and define a function $f_\mathbf{S}$ mapping intervals in $\mathbb{I}([0,1])$ to vertices in $\mathbf{S}$.

Definition 33. *Let $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ be a $\mathrm{D}_\sqsubset$-structure. The function $f_\mathbf{S} : \mathbb{I}([0,1]) \mapsto V$ is defined recursively as follows. First, $f_\mathbf{S}([0,1]) = v_0$. Now, let $[b, e]$ be an interval such that $f_\mathbf{S}([b, e]) = v$ and $f_\mathbf{S}$ has not been yet defined over any of its subinterval. We distinguish two cases.*

Case 1: *v is an irreflexive vertex. Let $\mathcal{C}_b$ and $\mathcal{C}_e$ be the reflexive successor beginning and ending clusters of v, respectively, and v_c be their common reflexive successor. Let v_b^{max} be the irreflexive successor of $\mathcal{C}_b$ (if any), v_e^{max} be the irreflexive successor of $\mathcal{C}_e$ (if any), and $v_1, \ldots, v_k$ be the k irreflexive successors of v_c (if any). Let $p = \frac{e-b}{2k+3}$. The function $f_\mathbf{S}$ is defined as follows:*

1. *we put $f_\mathbf{S}([b, b+p]) = v_b^{max}$ and $f_\mathbf{S}([e-p, e]) = v_e^{max}$;*
2. *for every $i = 1, \ldots, k$, we put $f_\mathbf{S}([b + 2ip, b + (2i+1)p]) = v_i$;*
3. *for every $i = 1, \ldots, k+1$, we put $f_\mathbf{S}([b + (2i-1)p, b + 2ip]) = v_c$;*

4. *for every* strict *subinterval* $[b', e']$ *of* $[b, e]$ *which is not a subinterval of any of the intervals* $[b + ip, b + (i + 1)p]$*, we put* $f_{\mathbf{S}}([b', e']) = v_c$*.*

To complete the construction, we need to define $f_{\mathbf{S}}$ *over the beginning subintervals* $[b, e']$ *such that* $b + p < e' < e$ *and the ending subintervals* $[b', e]$ *such that* $b < b' < e - p$*. We map such beginning (resp., ending) subintervals to vertices in* $\mathcal{C}_b$ *(resp.,* $\mathcal{C}_e$*) in such a way that for any beginning subinterval* $[b, e']$ *(resp., ending subinterval* $[b', e]$*) and any* $v_b \in \mathcal{C}_b$ *(resp.,* $v_e \in \mathcal{C}_e$*), there exists a beginning subinterval* $[b, e'']$*, with* $[b, b+p] \sqsubset [b, e''] \sqsubset [b, e']$ *(resp., ending subinterval* $[b'', e]$*, with* $[e-p, e] \sqsubset [b'', e] \sqsubset [b', e]$*) such that* $f_{\mathbf{S}}([b, e'']) = v_b$ *(resp.,* $f_{\mathbf{S}}([b'', e]) = v_e$*)*[1]*.*

Case 2: *v is a reflexive vertex. The case in which v belongs to $\mathcal{C}_b$ or $\mathcal{C}_e$ has been already dealt with. Thus, we only need to consider the case of vertices v_c with irreflexive successors only (apart from themselves). We distinguish two cases:*

1. v_c has no successors apart from itself. *In such a case, we put* $f_{\mathbf{S}}([b', e']) = v_c$ *for every subinterval* $[b', e']$ *of* $[b, e]$*.*
2. v_c has at least one successor different from itself. *Let* $v_c^1, \ldots, v_c^k$ *be the* k *successors of* v_c *different from* v_c*. We consider the intervals defined by the points* $b, b + p, b + 2p \ldots, b + 2kp, b + (2k + 1)p = e$*, with* $p = \frac{e-b}{2k+1}$*. The function* $f_{\mathbf{S}}$ *over such intervals is defined as follows:*
   - *for every* $i = 1, \ldots, k$*, we put* $f_{\mathbf{S}}([b + (2i - 1)p, b + 2ip]) = v_c^i$*.*
   - *for every* $i = 0, \ldots, k$*, we put* $f_{\mathbf{S}}([b + 2ip, b + (2i + 1)p]) = v_c$*.*

   We complete the construction by putting $f_{\mathbf{S}}([b', e']) = v_c$ *for every subinterval* $[b', e']$ *of* $[b, e]$ *which is not a subinterval of any of the intervals* $[b + ip, b + (i + 1)p]$*.*

The function $f_{\mathbf{S}}$ satisfies some basic properties.

Lemma 5.
1. *For every interval* $[b, e] \in \mathbb{I}([0, 1])$*, if* $f_{\mathbf{S}}([b, e]) = v$ *and* v' *is reachable from* v*, then there exists an interval* $[b', e']$ *such that* $f_{\mathbf{S}}([b', e']) = v'$ *and* $[b', e'] \sqsubset [b, e]$*.*
2. *For every pair of intervals* $[b, e]$ *and* $[b', e']$ *in* $\mathbb{I}([0, 1])$ *such that* $[b', e'] \sqsubset [b, e]$*, we have that for every formula* $[D]\psi \in \mathcal{L}(f_{\mathbf{S}}([b, e]))$*, both* ψ *and* $[D]\psi$ *belong to* $\mathcal{L}(f_{\mathbf{S}}([b', e']))$*.*

Proof. Condition 1 can be easily proved by observing that it trivially holds for all successors of v by definition of $f_{\mathbf{S}}$ and then extending the result to every descendant v' of v by induction on the length of the shortest path from v to v'.
As for condition 2, let $[b, e]$ and $[b', e']$ be two intervals in $\mathbb{I}([0, 1])$ such that $[b', e'] \sqsubset [b, e]$, $v = f_{\mathbf{S}}([b, e])$, and $v' = f_{\mathbf{S}}([b', e'])$. If v' is a descendant of v in the $\mathrm{D}_{\sqsubset}$-graph, then condition 2 holds by definition of D_φ. When we apply the construction step defined by Case 1, Point 4, of Definition 33, it may happen that $[b', e'] \sqsubset [b, e]$ but v' is not reachable from v in the $\mathrm{D}_{\sqsubset}$-graph. In such a case, both $[b, e]$ and $[b', e']$ are internal subintervals, and thus, by definition of the labeling functions $\mathcal{B}$ and $\mathcal{E}$, condition 2 is satisfied.

Theorem 39. *Given a fulfilling* $\mathrm{D}_{\sqsubset}$*-structure* $\mathbf{S}$ *for* φ*, there exists an interval model* $\boldsymbol{M}_{\mathbf{S}} = \langle \mathbb{I}([0, 1]), \sqsubset, \mathcal{V} \rangle$ *over the rational interval* $[0, 1]$ *such that* $\boldsymbol{M}_{\mathbf{S}}, [0, 1] \Vdash \varphi$*.*

Proof. For every $p \in \mathcal{AP}$, let $\mathcal{V}(p) = \{[b, e] : p \in \mathcal{L}(f_{\mathbf{S}}([b, e]))\}$. We can prove by induction on the structure of formulas $\psi \in \mathrm{CL}(\varphi)$ that for every interval $[b, e] \in \mathbb{I}([0, 1])$:

$$\mathbf{M}_{\mathbf{S}}, [b, e] \models \psi \text{ iff } \psi \in \mathcal{L}(f_{\mathbf{S}}([b, e])).$$

[1] Notice that the density of the rational interval $[0, 1]$ plays here an essential role.

The atomic case immediately follows from definition of $\mathcal{V}$; the Boolean cases follow from the definition of atom; finally, the case of temporal formulas follows from Lemma 5. This allows us to conclude that $\mathbf{M_S}, [0,1] \models \varphi$.

2.3 A small-model theorem for $\mathrm{D}_{\sqsubset}$-structures

Given a fulfilling $\mathrm{D}_{\sqsubset}$-structure, we can remove from it those vertices which are not necessary to fulfill any $\langle D \rangle$-formula to obtain a smaller $\mathrm{D}_{\sqsubset}$-structure of bounded size, as proved by the following theorem.

Theorem 40. *For every satisfiable $\mathrm{D}_{\sqsubset}$-formula φ, there exists a fulfilling $\mathrm{D}_{\sqsubset}$-structure with breadth and depth bounded by $2 \cdot |\varphi|$.*

Proof. Consider a fulfilling $\mathrm{D}_{\sqsubset}$-structure $\mathbf{S}$. The size of the structure can be safely reduced as follows:

- we remove from every cluster $\mathcal{C}$ all vertices that either do not fulfill any $\langle D \rangle$-formula or fulfill only formulas that are fulfilled by some descendant of it. Let $\mathcal{C}$ be the resulting cluster. We select a minimal subset $\mathcal{C}' \subseteq \mathcal{C}$ that fulfills all formulas that are fulfilled only inside $\mathcal{C}$ and we replace $\mathcal{C}$ with $\mathcal{C}'$ (if $\mathcal{C}'$ is empty, we replace $\mathcal{C}$ with one of its vertices);
- for every common reflexive successor v_c of a pair of clusters, we select a minimal subset of its irreflexive successors whose vertices satisfy all $\langle D \rangle$-formulas in v_c.

The execution of the first removal process produces a $\mathrm{D}_{\sqsubset}$-structure where the size of every cluster is at most $|\varphi|$ and every vertex in a cluster of size at least 2 fulfills some ψ formulas which are not fulfilled elsewhere, while the execution of the second removal process produces a $\mathrm{D}_{\sqsubset}$-structure where every vertex has at most $|\varphi|$ immediate successors.
Since whenever we exit from a cluster or we move from a reflexive node to an irreflexive one the number of requests strictly decreases, we can conclude that the length of every loop-free path is at most $2 \cdot |\varphi|$.

As a direct consequence of Theorem 40, we have that a fulfilling $\mathrm{D}_{\sqsubset}$-structure for a formula φ (if any) can be generated and explored by a non-deterministic procedure that uses only a polynomial amount of space. This gives the following complexity bound to the decision problem for $\mathrm{D}_{\sqsubset}$.

Theorem 41. *The decision problem for $\mathrm{D}_{\sqsubset}$ is in $PSPACE$.*

The very same reduction that has been used to prove $\mathrm{D}_{\sqsubseteq}$ PSPACE hardness in [13] can be applied to $\mathrm{D}_{\sqsubset}$, thus proving the PSPACE completeness of the satisfiability problem for $\mathrm{D}_{\sqsubset}$.

3 The tableau method for $\mathrm{D}_{\sqsubset}$

In this section we present a tableau system for $\mathrm{D}_{\sqsubset}$. From the model-theoretic results in the previous section, we have that a $\mathrm{D}_{\sqsubset}$-formula φ is satisfiable if and only if there exists a fulfilling $\mathrm{D}_{\sqsubset}$-structure for it. The tableau method attempts systematically to build such a structure if there is any, returning "satisfiable" if it succeeds and "unsatisfiable" otherwise. The nodes of the tableau are sets of locally consistent formulas (i.e., parts of atoms). At the root of the tableau, we place a set containing only the formula φ the satisfiability of which is being tested. We then proceed recursively to expand the tableau, following the expansion rules described below. Every disjunctive branch of the tableau describes an attempt to construct a fulfilling $D_{\sqsubset}$-structure for the atom at the root. Going down the branch roughly corresponds to going deeper into subintervals of the interval corresponding to the root. The applicability of an expansion rule at a given node depends on the formulas in the node and on the part of $D_{\sqsubset}$-structure we are building. The expansion of the tableau proceeds as follows.

1. We start with the *current vertex* (at the beginning, the root) v_0 of the $D_{\sqsubset}$-structure that is being constructed and we apply the usual Boolean rules to decompose Boolean operators.
2. Then, we impose a suitable marking on $\langle D\rangle$-formulas to partition them into four sets: the set of formulas that are satisfied only on beginning subintervals, that of formulas that are satisfied only on ending subintervals, that of formulas that are satisfied both on beginning and ending subintervals, and that of formulas that are satisfied on internal subintervals.
3. The third phase of the procedure is the construction of the first vertex v_b of the beginning successor cluster $\mathcal{C}_b$, the first vertex v_e of the ending successor cluster $\mathcal{C}_e$, and their common successor v_c.
4. Next, we proceed in parallel with the construction of the clusters $\mathcal{C}_b$ and $\mathcal{C}_e$ by guessing the $\langle D\rangle$-formulas from the set $REQ(\mathcal{L}(v_0))$ that should be satisfied inside each of them.
5. Then, we build the irreflexive successor v_b^{max} of $\mathcal{C}_b$, the irreflexive successor v_e^{max} of $\mathcal{C}_e$, and the irreflexive successors of v_c, if needed, and proceed recursively with their expansion from Step 1 above.

During the expansion of the tableau, we restrict our search to models with the property stated in Theorem 40. In particular, during the construction of a cluster we explicitly satisfy only those $\langle D\rangle$-formulas that should be satisfied inside the cluster and can never be satisfied outside it. In this way we have the following advantages:

i) we consider a $\langle D\rangle$-formula only once on a given branch of the tableau.

ii) when we exit a cluster, we can add the negation of every $\langle D\rangle$-formula that has been explicitly satisfied inside that cluster, thus reducing the search space of the successive expansion steps.

3.1 The rules of the tableau.

Before describing the tableau rules in details, we need to introduce some preliminary notation. A formula of the form $\langle D\rangle\psi \in CL(\varphi)$ can be possibly marked as follows:

$$\langle D\rangle^M\psi, \langle D\rangle^B\psi, \langle D\rangle^{BC}\psi, \langle D\rangle^{BNC}\psi, \langle D\rangle^E\psi, \langle D\rangle^{EC}\psi, \langle D\rangle^{ENC}\psi, \langle D\rangle^{BE}\psi.$$

This notation has the following intuitive meaning. The markings $\langle D\rangle^M\psi$, $\langle D\rangle^B\psi$, $\langle D\rangle^E\psi$, and $\langle D\rangle^{BE}$ appear when we try to construct an irreflexive interval node and we guess that the formula $\langle D\rangle\psi$ should be satisfied over an internal (middle) subinterval, only over a beginning subinterval, only over an ending subinterval, or both over a beginning and over an ending (but not over middle) subinterval of the current one. The markings $\langle D\rangle^{BC}\psi$ or $\langle D\rangle^{BNC}\psi$ (resp. $\langle D\rangle^{EC}\psi$,$\langle D\rangle^{ENC}\psi$) substitute a previously marked $\langle D\rangle^B\psi$ (resp. $\langle D\rangle^E\psi$) formula when we try to construct a beginning cluster and we guess that the formula ψ should be satisfied in the current cluster ($\langle D\rangle^{BC}\psi$ marking) or not ($\langle D\rangle^{BNC}\psi$ marking). The marking is only used for bookkeeping purposes, to facilitate the correct choice of the

rules to be applied. It does not affect the existence of a contradiction; we say that a *node is closed* iff once we remove the marking from every formula in it, it then contains both ψ and $\neg\psi$ for some $\psi \in CL(\varphi)$.
Given a set Φ of possibly marked formulas, the set $TF(\Phi)$ (the *temporal fragment of* Φ) is the set of all the formulas in Φ of the types $\langle D\rangle\psi$ and $[D]\psi$ (ignoring the markings). Given a set of formulas Γ, we use $(D)\Gamma$, where $(D) \in \{[D], \langle D\rangle, \langle D\rangle^M, \langle D\rangle^B, \langle D\rangle^{BC}, \langle D\rangle^{BNC}, \langle D\rangle^E, \langle D\rangle^{EC}, \langle D\rangle^{ENC}, \langle D\rangle^{BE}\}$, as a shorthand for $\{(D)\psi \mid \psi \in \Gamma\}$. Likewise, $\neg\Gamma$ stands for $\{\neg\psi \mid \psi \in \Gamma\}$ and $\Gamma \vee (D)\Gamma$ for $\{\psi \vee (D)\psi \mid \psi \in \Gamma\}$.

We now describe the rules used to expand the tableau nodes. In order to help the reader in understanding them, they are introduced and briefly explained in the order they appear in the procedure. We start with an initial tableau consisting of only one node containing the formula φ that we want to check for satisfiability. We apply the following **Boolean Rules** to $\{\varphi\}$ and to the newly generated nodes until these rules are no longer applicable:

$$\frac{\Phi, \neg\neg\psi}{\Phi, \psi} \qquad \frac{\Phi, \psi_1 \vee \psi_2}{\Phi, \psi_1 \mid \Phi, \psi_1} \qquad \frac{\Phi, \neg(\psi_1 \vee \psi_2)}{\Phi, \neg\psi_1, \neg\psi_2}$$

Next, we focus on a node to which the Boolean Rules are no more applicable. At this stage the node contains only atomic formulas and a subset of the temporal fragment of an atom (there may exist a formula $\langle D\rangle\psi \in REQ(\varphi)$ for which neither $\langle D\rangle\psi$ nor $[D]\neg\psi$ belongs to the current node). In order to obtain a complete temporal fragment, we apply the following **Completion Rule** to the current node and to all newly generated nodes:

$$\frac{\Phi}{\Phi, \langle D\rangle\psi \mid \Phi, [D]\neg\psi} \text{ where } \langle D\rangle\psi \in CL(\varphi),\ \langle D\rangle\psi \notin \Phi, \text{ and } [D]\neg\psi \notin \Phi.$$

Given a node with a complete temporal fragment, we have to classify every formula of the form $\langle D\rangle\psi$ belonging to it as a *beginning, middle, ending*, or *both beginning and ending* one. This is done by the following **Marking Rule**:

$$\frac{\Phi, \langle D\rangle\psi}{\Phi, \langle D\rangle^B\psi \mid \Phi, \langle D\rangle^M\psi \mid \Phi, \langle D\rangle^E\psi \mid \Phi, \langle D\rangle^{BE}\psi} \quad \text{where neither } \langle D\rangle^B\psi \text{ nor } \langle D\rangle^E\psi \text{ belongs to an ancestor of the current node.}$$

The conditions for the application of this rule will be explained later.

Given an irreflexive node with a complete temporal fragment, whose $\langle D\rangle$-formulas have been classified and marked, we generate its two reflexive successors, together with their common reflexive successor. This operation is performed by applying once the following **Reflexive Step Rule**:

$$\frac{\Phi, \langle D\rangle^B\Gamma, \langle D\rangle^M\mathrm{M}, \langle D\rangle^{BE}\Theta, \langle D\rangle^E\Lambda, [D]\Delta}{\begin{array}{c|c|c} \langle D\rangle^B\Gamma, \langle D\rangle^B\Theta, \langle D\rangle^M\mathrm{M}, & \langle D\rangle^M\mathrm{M}, & \langle D\rangle^E\Lambda, \langle D\rangle^E\Theta, \langle D\rangle^M\mathrm{M}, \\ [D]\neg\Lambda, [D]\Delta, \neg\Lambda, \Delta & [D]\neg\Gamma, [D]\neg\Theta, [D]\neg\Lambda, & [D]\neg\Gamma, [D]\Delta, \neg\Gamma, \Delta \\ & [D]\Delta, \neg\Gamma, \neg\Theta, \neg\Lambda, \Delta & \end{array}}$$

This rule splits the requests over three nodes accordingly to their classification. If a request cannot appear in a node, it introduces the corresponding negation. The generated nodes have a complete temporal fragment and are reflexive since all box arguments belong to them.

Now we have to deal with the expansion of the middle node. First, we apply the Boolean Rules until they are no longer applicable. Then, we apply the following **Middle Step Rule**:

$$\frac{\Phi, \langle D\rangle^M \mu_1, ..., \langle D\rangle^M \mu_h, [D]\Gamma}{\mu_1, \Gamma, [D]\Gamma \big| ... \big| \mu_h, \Gamma, [D]\Gamma}$$

For every request in the current node, this rule creates an irreflexive successor of it. Then, we re-apply the expansion procedure from the beginning for every newly generated node.

The expansion of a beginning node takes place as follows. As usual, we first apply the Boolean Rules to it, and to the newly generated nodes, until they are applicable. Then, for any $\langle D\rangle^B \psi$ formula in the current node, we distinguish two cases: $\langle D\rangle^B \psi$ can be fulfilled in the cluster or it can be fulfilled in one of its descendants. They are dealt with the following **Build Beginning Cluster Rule**:

$$\frac{\Phi, \langle D\rangle^B \psi, \langle D\rangle^B \Gamma_B, \langle D\rangle^{BC} \Gamma_{BC}, \langle D\rangle^{BNC} \Gamma_{BNC}, \langle D\rangle^M \mathrm{M}, [D]\Delta}{\begin{array}{c|c} \psi, \langle D\rangle^B \Gamma_B, \langle D\rangle^{BC}(\Gamma_{BC} \cup \{\psi\}), & \Phi, \langle D\rangle^B \Gamma_B, \langle D\rangle^{BC} \Gamma_{BC}, \\ \langle D\rangle^{BNC} \Gamma_{BNC}, \langle D\rangle^M \mathrm{M}, [D]\Delta, \Delta & \langle D\rangle^{BNC}(\Gamma_{BNC} \cup \{\psi\}), \langle D\rangle^M \mathrm{M}, [D]\Delta \end{array}}$$

The former case is handled by the first branch, which marks the request as $\langle D\rangle^{BC} \psi$ (in order to avoid loops) and satisfies ψ in a new cluster node with the same temporal fragment as the current one. The latter case is handled by the second branch that simply reclassifies the request as $\langle D\rangle^{BNC} \psi$ without moving to another cluster node. Such a procedure is iterated until every $\langle D\rangle^B \psi$ is re-marked as $\langle D\rangle^{BC} \psi$ or $\langle D\rangle^{BNC} \psi$.

The case of ending nodes is dealt with in a very similar way by means of the following **Build Ending Cluster Rule**:

$$\frac{\Phi, \langle D\rangle^E \psi, \langle D\rangle^E \Gamma_E, \langle D\rangle^{EC} \Gamma_{EC}, \langle D\rangle^{ENC} \Gamma_{ENC}, \langle D\rangle^M \mathrm{M}, [D]\Delta}{\begin{array}{c|c} \psi, \langle D\rangle^E \Gamma_E, \langle D\rangle^{EC}(\Gamma_{EC} \cup \{\psi\}), & \Phi, \langle D\rangle^E \Gamma_E, \langle D\rangle^{EC} \Gamma_{EC}, \\ \langle D\rangle^{ENC} \Gamma_{ENC}, \langle D\rangle^M \mathrm{M}, [D]\Delta, \Delta & \langle D\rangle^{ENC}(\Gamma_{ENC} \cup \{\psi\}), \langle D\rangle^M \mathrm{M}, [D]\Delta \end{array}}$$

Once we reach a cluster node such that no Boolean rules are applicable and every $\langle D\rangle^B \psi$ request has been reclassified as $\langle D\rangle^{BC} \psi$ or $\langle D\rangle^{BNC} \psi$, we proceed as follows. If the node does not include any $\langle D\rangle^{BNC} \psi$ request, we are done (all requests have been satisfied in the cluster). Otherwise (there exists at least one marked formula of the form $\langle D\rangle^{BNC} \psi$), we generate an irreflexive successor of the cluster that, for every formula $\langle D\rangle^{BNC} \psi$, satisfies either ψ or $\langle D\rangle^B \psi$. This last case is handled by the formulas $\Gamma_{BNC} \vee \langle D\rangle^B \Gamma_{BNC}$ introduced by the following **Exit Beginning Cluster Rule**:

$$\frac{\Phi, \langle D\rangle^{BC} \Gamma_{BC}, \langle D\rangle^{BNC} \Gamma_{BNC}, \langle D\rangle^M \mathrm{M}, [D]\Delta}{\Gamma_{BNC} \vee \langle D\rangle^B \Gamma_{BNC}, [D]\neg \Gamma_{BC}, [D]\Delta, \Delta} \text{ where } \Gamma_{BNC} \neq \emptyset.$$

The case of the ending cluster is dealt with in a very similar way by means of the following **Exit Ending Cluster Rule**:

$$\frac{\Phi, \langle D\rangle^{EC} \Gamma_{EC}, \langle D\rangle^{ENC} \Gamma_{ENC}, \langle D\rangle^M \mathrm{M}, [D]\Delta}{\Gamma_{ENC} \vee \langle D\rangle^E \Gamma_{ENC}, [D]\neg \Gamma_{EC}, [D]\Delta, \Delta} \text{ where } \Gamma_{ENC} \neq \emptyset.$$

Then, we apply again all steps from the beginning, with only a little difference in the application of the Marking Rule. The Completion Rule may produce some requests $\langle D\rangle\psi$

devoid of any markings. For all these requests, we must check whether they have been marked as $\langle D\rangle^B\psi$ or $\langle D\rangle^E\psi$ in an ancestor of the current node and, if this is the case, we must guarantee the downward propagation of their markings. To this end, before applying the Marking Rule, we apply the following **Persistent Beginning** and **Persistent Ending Rules**:

$$\frac{\Phi, \langle D\rangle\psi}{\Phi, \langle D\rangle^B\psi} \qquad \frac{\Phi, \langle D\rangle\psi}{\Phi, \langle D\rangle^E\psi}$$

whenever $\langle D\rangle^B\psi$ (resp., $\langle D\rangle^E\psi$) belongs to an ancestor of the current node.

3.2 Building the tableaux.

A tableau for a $\mathrm{D}_{\sqsubset}$-formula φ is a finite graph $\mathcal{T} = \langle V, E\rangle$, whose vertices are subsets of $\mathrm{CL}(\varphi)$ and whose edges are generated by the application of expansion rules. The construction of the tableau starts with the *initial tableau*, which is the single node graph $\langle\{\{\varphi\}\}, \emptyset\rangle$. To describe such a construction process, we take advantage of macronodes, which can be viewed as the counterpart of vertices of $\mathrm{D}_{\sqsubset}$-structures.
Given a set $V' \subseteq V$, let $E(V')$ be the restriction of E to vertices in V. Moreover, let the Reflexive Step, Middle Step, Build Beginning/Ending Cluster and Exit Beginning/Ending Cluster rules be called **Step Rules**. Macronodes are defined as follows.

Definition 34. *Let $\langle V, E\rangle$ be a tableau for a $\mathrm{D}_{\sqsubset}$-formula φ. A* macronode *is a set $V' \subseteq V$ such that:*

- *$\langle V', E(V')\rangle$ is a tree;*
- *the root of $\langle V', E(V')\rangle$ is either the initial node of the tableau or a node generated by an application of a Step Rule;*
- *every edge in $E(V')$ is generated by the application of an expansion rule which is not a Step Rule;*
- *the only expansion rule that can be applied to the leaves of $\langle V', E(V')\rangle$ is a Step Rule.*

A macronode m is reflexive *if its root is generated by the application of the Reflexive Step Rule or of the Build Beginning/Ending Cluster Rules; otherwise, it is* irreflexive.

We say that a rule is applicable to a node n if it generates at least one successor node. The construction of a tableau for a $\mathrm{D}_{\sqsubset}$-formula φ starts with the initial tableau $\langle\{\{\varphi\}\}, \emptyset\rangle$ and proceeds by applying the following *expansion strategy* to the leaves of the current tableau, until it cannot be applied anymore.

Apply the first rule in the list whose condition is satisfied:

1. a Boolean Rule is applicable;
2. the Completion Rule is applicable;
3. the node belongs to an irreflexive macronode and the Persistent Beginning Rule is applicable;
4. the node belongs to an irreflexive macronode and the Persistent Ending Rule is applicable;
5. the node belongs to an irreflexive macronode and the Marking Rule is applicable;
6. the node belongs to an irreflexive macronode and the Reflexive Step Rule is applicable;

7. the node belongs to a reflexive macronode with only M markings and the Middle Step Rule is applicable;
8. the node belongs to a reflexive macronode with B markings or E markings and the Build Beginning/Ending Cluster Rules are applicable;
9. the node belongs to a reflexive macronode with B markings or E markings and the Exit Beginning/Ending Cluster Rules are applicable.

Termination is ensured by the following *looping conditions*:

– if an application of the Reflexive Rule generates a node which is the root of an existing reflexive macronode, then add an edge from the current node to this node instead of creating the new one.
– if the Middle Step Rule is applied to a node n and one of the successor nodes it generates, say n', is such that $TF(n') = TF(n)$, then add the edge (n', n) to the tableau. Do not apply any expansion rule to n'.

We say that a node n in a tableau is *closed* if one of the following conditions holds:

– there exists ψ such that both ψ and $\neg\psi$ belong to n;
– a Middle Step Rule or a Reflexive Step Rule have been applied to n and *at least one* of its successors is closed;
– a rule different from the Middle Step Rule and the Reflexive Step Rule has been applied to n and *all* its successors are closed;
– n is a descendant of a node n' to which an Exit Beginning/Ending Cluster Rule has been applied and $TF(n') = TF(n)$.

A node in a tableau is *open* if it is not closed. A tableau is *open* if and only if its root is open. We will prove that a formula is satisfiable if and only if there exists an open tableau for it.
As for computational complexity, it is not difficult to show that the proof of Theorem 40 can be adapted to the proposed tableau method. The only difference is that at any step of the tableau construction we either expand a node or mark one of its formulas. As a consequence, any node of a $D_{\sqsubset}$-structure corresponds to a path of at most $|\varphi|$ nodes in the tableau. Hence, the depth of the tableau is bounded by $2 \cdot |\varphi|^2$. Since the breadth of the tableau is $2 \cdot |\varphi|$, we can conclude that the proposed tableau-based decision procedure is in $PSPACE$ (and thus optimal).

Theorem 42. *(Complexity) The proposed tableau procedure is in $PSPACE$.*

3.3 Example of application.

Here we give an example of the above-described expansion strategy at work. Consider the formula $\varphi = \langle D \rangle p \wedge \langle D \rangle q \wedge [D]\neg(\langle D \rangle p \wedge \langle D \rangle q)$, which states that the given interval has a subinterval where p holds and a subinterval where q holds, but no subintervals covering both of them. It is easy to see that in any model for this formula p and q respectively hold in a beginning and an ending subinterval only, or vice versa. Part of the tableau for φ is depicted in Figure 6. Due to space limitations, we restrict our attention to the non-closed region of the tableau and we skip the details about the application of Boolean Rules. We start with the root A, whose temporal fragment is complete, and we apply the Marking Rule. For the sake of conciseness, we only consider a correct marking, which inserts $\langle D\rangle^B p$ and $\langle D\rangle^E q$ in B. Once all $\langle D\rangle$-formulas have been marked, we apply the Reflexive Step Rule, that generates the three successors of B. The first successor is node C that contains the request $\langle D\rangle^B p$ and the negation of the request $\langle D\rangle^E q$, namely, $[D]\neg q$. The second one is node E that contains the request $\langle D\rangle^E q$ and the negation of the request $\langle D\rangle^B p$, namely, $[D]\neg p$. The third one is node D that contains the negation of the two requests (such a node represents the middle reflexive vertex of the corresponding $D_{\sqsubset}$-structure). Node D contains no $\langle D\rangle$-formulas and thus it cannot be expanded anymore. Since it does not include any contradiction, we declare it open. Consider now node C. According to the expansion strategy, we apply the Build Beginning Cluster Rule to $\langle D\rangle^B p$ in node C, that generates nodes F and G. Node F includes p and, accordingly, replaces $\langle D\rangle^B p$ with $\langle D\rangle^{BC} p$. It does not contain $\langle D\rangle^{BNC}$ formulas and no expansion rules are applicable to it. Since it does not include any contradiction, we declare it open. The same argument can be applied to nodes E and H. This allows us to conclude that the tableau is open (and thus φ is satisfiable).

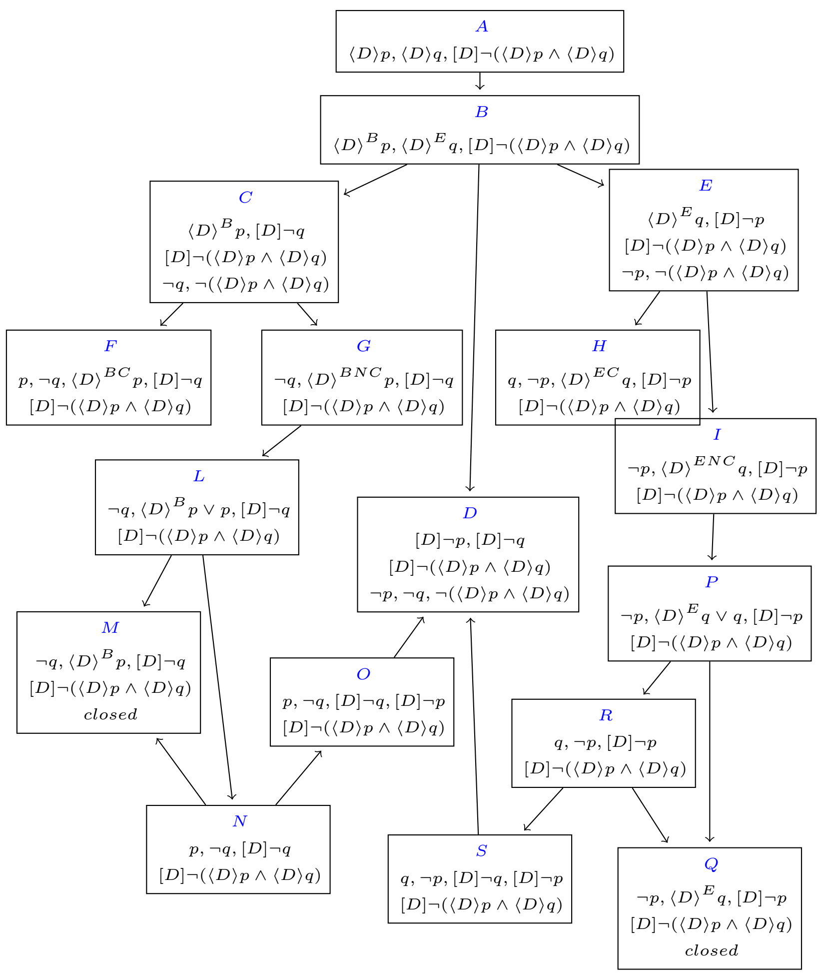


Fig. 6. (Part of) the tableau for $\varphi = \langle D\rangle p \wedge \langle D\rangle q \wedge [D]\neg(\langle D\rangle p \wedge \langle D\rangle q)$.

To better explain the proposed tableau method, we include in Figure 6 additional nodes which are not strictly necessary to conclude that the tableau is open. This is the case with node G that replaces $\langle D\rangle^B p$ with $\langle D\rangle^{BNC} p$, thus postponing the satisfaction of p. According to the expansion strategy, we apply the Exit Beginning Cluster Rule to G, that generates the irreflexive node L. Such a node contains the formula $\langle D\rangle^B p \vee p$, stating that p is satisfied either in L or in some descendant of it. The application of the Or Rule to $\langle D\rangle^B p \vee p$ generates nodes M and N. Node M includes again the formula $\langle D\rangle^B p$ and, since $TF(M) = TF(G)$, we declare it closed. As for node N, that satisfies p, we apply the Completion Rule (neither $\langle D\rangle p$ nor $[D]\neg p$ belongs to N), that generates its two successors. The first successor turns out to be identical to M and thus we add an edge from N to M instead of adding a new node; the second successor is node O, with $TF(O) \subset TF(G)$. Then, we apply Reflexive Step Rule to node O. Since it does not contain any $\langle D\rangle$-formula, its three reflexive successors coincides with node D. Hence, we add an edge from O to D and we stop the expansion of (this part of) the tableau.

3.4 Soundness and completeness

We conclude the section by proving soundness and completeness of the tableau method.

Theorem 43. *(SOUNDNESS) Let φ be a $\mathrm{D}_\sqsubset$-formula and $\mathcal{T}$ be a tableau for it. If $\mathcal{T}$ is open, then φ is satisfiable.*

Proof. We build a fulfilling $\mathrm{D}_\sqsubset$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ for φ step by step, starting from the root of $\mathcal{T}$ and proceeding according to the expansion rules that have been applied in the construction of the tableau.

Let n_0 be the root of $\mathcal{T}$. We generate the one-node $\mathrm{D}_\sqsubset$-graph $\langle\{v_0\}, \emptyset\rangle$ and we put formulas belonging to n_0 in $\mathcal{L}(v_0)$. Now, let n be an open node in $\mathcal{T}$ and let v be the corresponding vertex in the $\mathrm{D}_\sqsubset$-graph. The way in which we develop the $\mathrm{D}_\sqsubset$-structure depends on the expansion rule that has been applied to n during the construction of the tableau.

- *A Boolean Rule has been applied.* Then, at least one successor n' of n is open. We add formulas belonging to n' to $\mathcal{L}(v)$ and we proceed by taking into consideration the tableau node n' and the vertex v.
- *The Completion Rule has been applied.* Then, at least one successor n' of n is open. As in the previous case, we add formulas belonging to n' to $\mathcal{L}(v)$ and we proceed by taking into consideration the tableau node n' and the vertex v.
- *The Marking/Persistent Beginning/Persistent Ending Rule has been applied.* Let $\langle D\rangle\psi$ be the formula to which the rule has been applied and let n' be one of the open successors of n. Four cases may arise, depending on which marking has been applied to the considered formula in n':
  - if $\langle D\rangle^B\psi \in n'$, then we put $\langle D\rangle\psi \in \mathcal{B}(v)$;
  - if $\langle D\rangle^E\psi \in n'$, then we put $\langle D\rangle\psi \in \mathcal{E}(v)$;
  - if $\langle D\rangle^{BE}\psi \in n'$, then we add $\langle D\rangle\psi$ to both $\mathcal{B}(v)$ and $\mathcal{E}(v)$;
  - if $\langle D\rangle^M\psi \in n'$, then the marking does not influence the construction of the $\mathrm{D}_\sqsubset$-structure.

  In all cases, we proceed recursively by taking into consideration the tableau node n' and the current vertex v.
- *The Reflexive Step Rule has been applied.* Since $\mathcal{T}$ is open, all successors of n are open either. Let n_b, n_c, and n_e be the first, second, and third successor of n, respectively. We add three reflexive vertices v_b, v_c, and v_e to V and the edges (v, v_b), (v, v_e), (v_b, v_c), (v_e, v_c), (v_b, v_b), (v_c, v_c), and (v_e, v_e) to E. The labeling of v_b, v_c, and v_e is defined as follows: $\mathcal{L}(v_b) = n_b$, $\mathcal{L}(v_c) = n_c$, and $\mathcal{L}(v_e) = n_e$. We recursively apply the construction

by taking into consideration the node n_b with the corresponding vertex v_b, the node n_c with the corresponding vertex v_c, and the node n_e with the corresponding vertex v_e.

- *The Middle Step Rule has been applied.* Since n is open, all its successors $n_1, ..., n_h$ are open either. We add h new vertices $v_1, ..., v_h$ to V and the edges $(v, v_1), ..., (v, v_h)$ to E, and we define their labeling in such a way that for $i = 1, \ldots, h$, $\mathcal{L}(v_i) = n_i$. We recursively apply the construction to every node n_i paired with the corresponding vertex v_i.
- *The Build Beginning/Ending Cluster Rule has been applied.* Suppose that the rule has been applied to a formula $\langle D\rangle^B\psi \in n$ (the case of $\langle D\rangle^E\psi$ is analogous) and let n' be an open successor of n. Two cases may arise:
  1. $\langle D\rangle^{BC}\psi \in n'$ ($\langle D\rangle\psi$ has been satisfied in the cluster). We introduce a new node v' in the cluster of v by adding the edges $(v, v'), (v', v')$, and (v', v) to E. The labeling $\mathcal{L}(v')$ of v' consists of the set of formulas belonging to n'. We proceed by taking into consideration the node n' and the corresponding vertex v'.
  2. $\langle D\rangle^{BNC}\psi \in n'$ (satisfaction of $\langle D\rangle\psi$ has been postponed). We do not add any vertex to the $\mathrm{D}_{\sqsubset}$-structure, but simply proceed by taking into consideration the node n' and the current vertex v.
- *The Exit Beginning/Ending Cluster Rule has been applied.* Since $\mathcal{T}$ is open, the unique successor n' of n is open and it is the root of an irreflexive macronode. We add a new irreflexive vertex v' to V and an edge (v, v') to E. Moreover, we set the labeling of v' as the set of formulas belonging to n'. Then, we proceed by taking into consideration the node n' with the corresponding vertex v'.

To keep the construction finite, whenever the procedure reaches a tableau node n' that has been already taken into consideration, instead of adding a new vertex to the $\mathrm{D}_{\sqsubset}$-structure, it simply adds an edge from the current vertex v to the vertex v' corresponding to n'.
Since any tableau for φ is finite, such a construction is terminating. However, the resulting structure $\langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ is not necessarily a $\mathrm{D}_{\sqsubset}$-structure: there may exist a vertex $v \in V$ and a non-temporal formula $\psi \in \mathrm{CL}(\varphi)$ such that neither ψ nor $\neg\psi$ belongs to $\mathcal{L}(v)$. To overcome this problem, we can consistently extend the labeling $\mathcal{L}(v)$ as follows:

- if $\psi = p$, with $p \in \mathcal{AP}$, we put $\neg p \in \mathcal{L}(v)$;
- If $\psi = \neg\xi$, we put $\psi \in \mathcal{L}(v)$ if and only if $\xi \notin \mathcal{L}(v)$;
- If $\psi = \psi_1 \vee \psi_2$, we put $\psi_1 \vee \psi_2 \in \mathcal{L}(v)$ if and only if $\psi_1 \in \mathcal{L}(v)$ or $\psi_2 \in \mathcal{L}(v)$.

The resulting $\mathrm{D}_{\sqsubset}$-structure $\langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ is a fulfilling $\mathrm{D}_{\sqsubset}$-structure for φ and thus φ is satisfiable. □

Theorem 44. (COMPLETENESS) *Let φ be a $\mathrm{D}_{\sqsubset}$-formula. If φ is satisfiable, then there exists an open tableau for it.*

Proof. Let $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ be a fulfilling $\mathrm{D}_{\sqsubset}$-structure that satisfies φ. We take advantage of such a structure to show that there exists an open tableau $\mathcal{T}$ for φ. In particular, we will define a correspondence between (some) nodes in $\mathcal{T}$ and vertices in $\mathbf{S}$ that satisfies the following constraints:

(1) if n is associated with an irreflexive vertex v, then n belongs to an irreflexive macronode;
(2) if n is associated with a reflexive vertex v, then n belongs to a reflexive macronode;
(3) ff n is associated with a vertex v, then, for every formula $\psi \in n$, $\psi \in \mathcal{L}(v)$.

Let n_0 be the root of the tableau. We associate it with the root v_0 of $\mathbf{S}$. Since n_0 belongs to an irreflexive macronode, v_0 is an irreflexive vertex, and $\varphi \in \mathcal{L}(v_0)$, all constraints are satisfied.
Let n be the current node of the tableau, v be the vertex of $\mathbf{S}$ associated with it, and, by inductive hypothesis, n and v satisfy the constraints. We proceed by taking into consideration the rule that, according to the expansion strategy, is applicable to node n.

- *One of the Boolean Rules is applicable.* We consider the application of the OR Rule to a formula of the form $\psi_1 \vee \psi_2$ (the other cases are simpler and thus omitted). Since $\psi_1 \vee \psi_2 \in n$, by Constraint (3), $\psi_1 \vee \psi_2 \in \mathcal{L}(v)$ and thus $\psi_1 \in \mathcal{L}(v)$ or $\psi_2 \in \mathcal{L}(v)$. If $\psi_1 \in \mathcal{L}(v)$, then we associate the successor n_1 of n, that contains ψ_1, with v; otherwise, we associate the successor n_2 of n, that contains ψ_2, with v. In either cases, all constraints are satisfied.
- *The Completion Rule is applicable.* Let us consider the application of the Completion Rule to the formula $\langle D\rangle\psi$. Since $\mathcal{L}(v)$ is an atom, either $\langle D\rangle\psi \in \mathcal{L}(v)$ or $[D]\neg\psi \in \mathcal{L}(v)$. In the former case, we associate the successor n_1 of n, that contains $\langle D\rangle\psi$, with v; in the latter case, we associate the successor n_2 of n, containing $[D]\neg\psi$, with v. In either cases, all constraints are satisfied.
- *The Marking Rule is applicable.* Let us consider the application of the Marking Rule to the formula $\langle D\rangle\psi$. According to the expansion strategy, n belongs to an irreflexive macronode and thus, by inductive hypothesis, v is an irreflexive vertex. Let $\mathcal{C}_b$ be the beginning successor cluster of v, $\mathcal{C}_e$ the ending successor cluster of v, and v_c their common reflexive successor (see Definition 30). Four cases may arise:
  1. $\langle D\rangle\psi$ appears in $\mathcal{C}_b$, but not in $\mathcal{C}_e$ and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^B\psi$, with v.
  2. $\langle D\rangle\psi$ appears in $\mathcal{C}_e$, but not in $\mathcal{C}_b$ and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^E\psi$, with v.
  3. $\langle D\rangle\psi$ appears in $\mathcal{C}_b$ and $\mathcal{C}_e$, but not in v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^{BE}\psi$, with v.
  4. $\langle D\rangle\psi$ appears in $\mathcal{C}_b$, $\mathcal{C}_e$, and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^M\psi$, with v.
- *The Persistent Beginning/Ending Rule is applicable.* We associate the unique successor n' of n with v.
- *The Reflexive Step Rule is applicable.* According to the expansion strategy, n belongs to an irreflexive macronode and thus, by inductive hypothesis, v is an irreflexive vertex. Let v_b be a node in the beginning successor cluster of v, v_e a node in the ending successor cluster of v, and v_c the common reflexive successor of the two clusters. According to the expansion strategy, when such a rule turns out to be applicable, all $\langle D\rangle$-formulas have already been marked in accordance with $\mathbf{S}$. Let $n = \{\Phi, \langle D\rangle^B\Gamma, \langle D\rangle^M\mathrm{M}, \langle D\rangle^{BE}\Theta, \langle D\rangle^E\Lambda, [D]\Delta\}$, where Φ only contains atomic formulas. We have that $\{\langle D\rangle\Gamma, \langle D\rangle\Theta, \langle D\rangle\mathrm{M}, [D]\neg\Lambda, [D]\Delta, \neg\Lambda, \Delta\} \subseteq \mathcal{L}(v_b)$, that $\{\langle D\rangle\Lambda, \langle D\rangle\Theta, \langle D\rangle\mathrm{M}, [D]\neg\Gamma, [D]\Delta, \neg\Gamma, \Delta\} \subseteq \mathcal{L}(v_e)$, and that $\{\langle D\rangle\mathrm{M}, [D]\neg\Gamma, [D]\neg\Theta[D]\neg\Lambda, [D]\Delta, \neg\Gamma, \neg\Theta, \neg\Lambda, \Delta\} \subseteq \mathcal{L}(v_c)$. We associate the first successor of n with v_b, the second one with v_e, and the third one with v_c.
- *The Middle Step Rule is applicable.* According to the expansion strategy, n belongs to a macronode whose root is the middle node generated by an application of the Reflexive Step Rule and thus, by inductive hypothesis, n is associated with a middle reflexive vertex v_c. Since $\mathbf{S}$ is fulfilling, for every formula $\langle D\rangle\psi \in n$ there exists a successor v_ψ of

v_c such that $\psi \in \mathcal{L}(v_\psi)$ and for every $[D]\theta \in n$, $\theta, [D]\theta \in \mathcal{L}(v_\psi)$. For all $\langle D\rangle\psi \in n$, we associated the successor n_ψ of n with v_ψ.

- *The Build Beginning Cluster Rule is applicable.* Given the expansion strategy, by inductive hypothesis we have that n is associated with a node v that belongs to a beginning cluster $\mathcal{C}$. Let us consider the application of the rule to the formula $\langle D\rangle^B\psi$. Two cases may arise: either **S** fulfills $\langle D\rangle\psi$ outside $\mathcal{C}$ or not. In the former case, we associate the successor n' of n, that contains $\langle D\rangle^{BNC}\psi$, with v; in the latter case, there exists a node $v' \in \mathcal{C}$ such that $\psi \in \mathcal{L}(v')$ and we associate the successor n' of n, that contains both ψ and $\langle D\rangle^{BC}\psi$, with v'.
- *The Build Ending Cluster Rule is applicable.* This case is analogous to the previous one and thus omitted.
- *The Exit Beginning Cluster Rule is applicable.* Given the expansion strategy, by inductive hypothesis we have that n is associated with a node v that belongs to a beginning cluster $\mathcal{C}$. Let v' be the unique irreflexive successor of $\mathcal{C}$. We have that, for every formula $\langle D\rangle^{BNC}\psi \in n$, $\psi \in \mathcal{L}(v')$ or $\langle D\rangle\psi \in \mathcal{L}(v')$. The labeling of the unique successor node n' of n is thus consistent with v' and we can associate n' with v'.
- *The Exit Ending Cluster Rule is applicable.* This case is analogous to the previous one and thus omitted.

At the end of the above construction, we have obtained (a portion of) a tableau for φ. Since all its nodes are open, we can conclude that there exists an open tableau for φ. $\square$

4 Conclusions

In [13], we devised a technique for constructing finite pseudo-models and building tableau-based decision procedures for logics of subinterval structures and applied it to the logic of strict subintervals. In this paper, we generalized it to the much more difficult case of the logic of proper subintervals. In such a way, we have completed the analysis and the proof of decidability for all versions of the semantics of subinterval logics (strict, proper, and reflexive) over dense linear orders, where point-intervals are not admitted. The inclusion of point-intervals is, however, unproblematic, because in the two difficult cases (strict and proper subinterval semantics) they are definable over dense linear orders by the formula $\langle D\rangle\bot$. Thus, the decidability results and tableau constructions carry over to subinterval structures with point-intervals after suitable minor modifications. On the contrary, the cases of discrete and arbitrary linear orders seem rather more difficult, and they are currently still under investigation.

Copy of: **D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional Interval Neighborhood Logics: Expressiveness, Decidability, and Undecidable Extensions. Research Report 05, Department of Mathematics and Computer Science, University of Udine, Italy, February 2008**

Abstract. In this paper, we investigate the expressiveness of the variety of propositional interval neighborhood logics (PNL), we establish their decidability on linearly ordered domains and some important sub-classes, and we prove undecidability of a number of extensions of PNL with additional modalities over interval relations. Thus, we show that PNL form a quite expressive and nearly maximal decidable fragment of Halpern-Shoham's interval logic HS.

1 Introduction

The study of interval-based temporal logics on linearly ordered domains is an emerging research area of increasing importance in computer science and artificial intelligence. A recent survey of the main developments, results, and open problems in the area can be found in [52]. The main systems of propositional interval temporal logics studied so far include Moszkowski's Propositional Interval Logic (PITL) [84], Halpern and Shoham's modal logic of time intervals (HS) [59], and Venema's CDTlogic [110] (extended to branching-time frames with linear intervals in [53]). Important fragments of HSstudied in more detail include the logic of *begins/ends* (BE) [74], the logics of temporal neighborhood [20,21,50], and the logics of subinterval structures [12,13]. Unfortunately, even when restricted to the case of propositional languages and linear time, interval logics are usually undecidable. In particular, PITL was proved to be undecidable on the classes of discrete and finite frames in [84]; undecidability on dense linearly ordered sets was prove by Lodaya in [74]. Likewise, the logic HS(and therefore CDT) was shown to be (often highly) undecidable in most natural classes of frames in [59]. That result was sharpened by Lodaya in [74] where the undecidability of BE on dense orderings was proved (as noted in [52], this result carries over to the class of all linearly ordered sets).
Decidability results for interval logics are scarce; moreover, most of them are obtained by imposing various restrictions on the semantics, such as the projection principles of locality and homogeneity [59,84], or restrictions on the family of available intervals in the model, such as 'split structures' [82]. So far, very few unrestricted decidability results for fragments of HSare known, which are based on tableau methods, e.g., the NEXPTIME decision procedure for the future fragment of neighborhood logic interpreted on $\mathbb{N}$ [18,21], later extended to full neighborhood logic over $\mathbb{Z}$ [20], and the PSPACE decision procedures for the logics of strict and proper subinterval structures over dense linear orderings [12,13]
In this paper, we address expressiveness and decidability issues for propositional neighborhood logics (PNL). These are fragments of HSwhich feature the modalities corresponding to the relations of right-adjacent and left-adjacent intervals (in terms of Allen's relations, *meets/met by*), and (possibly) the modal constant π, which is true precisely on point-intervals (intervals with coinciding endpoints). We focus our attention on three variants of PNL, namely, PNL$^-$, based on strict semantics that excludes point-intervals, PNL$^+$, based on non-strict semantics that includes point-intervals, and PNL$^{\pi+}$, that extends PNL$^+$ with π. Besides the above-mentioned decidability results for $\mathbb{N}$ and $\mathbb{Z}$, a number of representation theorems and sound and complete axiomatic systems on various classes of linear orders, as well as a tableau-based semi-decision procedure, have been obtained for PNL [50].

The main results given in the present paper are:

1. NEXPTIME-complete decidability of the satisfiability problem for $\mathrm{PNL}^{\pi+}$ on some important classes of linear orders. This result hinges upon the decidability of the satisfiability problem for the two-variable fragment of first-order logic $\mathrm{FO}^2[<]$ for binary relational structures over ordered domains, due to Otto [92]. Thus, while the main technical work behind this result has already been done elsewhere, we emphasize here on its conceptual importance, being the first decidable general case of natural and expressive interval languages interpreted in genuine, unrestricted interval-based semantics.
2. Expressive completeness of $\mathrm{PNL}^{\pi+}$ with respect to $\mathrm{FO}^2[<]$, by means of a suitable faithful translation of the latter into the former. This result is in the spirit of seminal Kamp's theorem [70]. Kamp proved the functional completeness of the *Since* (S) and *Until* (U) temporal logic with respect to first-order definable connectives on Dedekind-complete linear orders. This result has been later re-proved and generalized in several ways (see [44,69]). In particular, Stavi extended Kamp's result to the class of all linear orders by adding the binary operators S' and U' (see [44] for details), while Etessami et al. [42] proved the functional completeness of the linear-time temporal logic with future and past operators F, P with respect to the two-variable, unary-predicate fragment of first-order logic over $\mathbb{N}$. Finally, Venema proved the expressive completeness of CDT with respect to the three-variable fragment of first-order logic with at most two free variables $\mathrm{FO}^3_{x,y}[<]$ on the class of all linear orders [110]. These expressive completeness results are important from both perspectives: for propositional interval logics and for bounded-variable fragments of first-order logic for relational structures over ordered domains, as they open the perspective for cross-fertilization of these fields, esp. with regards to decision procedures.
3. Undecidability of a number of extensions of $\mathrm{PNL}^{\pi+}$ with various additional interval modalities from the HSrepertoire. The technique used to obtain these results is a non-trivial reduction from the (undecidable) tiling problem for an octant of the integer plane. That technique is quite versatile and can be applied to a variety of extensions of $\mathrm{PNL}^{\pi+}$, thus showing that $\mathrm{PNL}^{\pi+}$ is very close to being a maximal decidable fragment of HS (as a matter of fact, we conjecture that it is a *maximal* fragment with that property).

The rest of the paper is organized as follows. After preliminaries, in Section 3, we compare the expressive power of PNL^-, PNL^+, and $\mathrm{PNL}^{\pi+}$. We show that $\mathrm{PNL}^{\pi+}$ is strictly more expressive than PNL^+ and PNL^-, while the latter two are incomparable in terms of expressiveness. Then, in Section 4 we prove the decidability of the satisfiability problem for $\mathrm{PNL}^{\pi+}$ on the classes of all linear orders, all well-orders, all finite linear orders, and $\mathbb{N}$, by reduction to Otto's results. Next, in Section 5 we provide a translation of $\mathrm{FO}^2[<]$ into $\mathrm{PNL}^{\pi+}$, thus proving expressive completeness of the latter with respect to the former on the class of all linear orders, while in Section 6 we show that $\mathrm{PNL}^{\pi+}$ is a maximal fragment of HSthat translates into $\mathrm{FO}^2[<]$. In Section 7, we establish undecidability of various extensions of $\mathrm{PNL}^{\pi+}$ with additional interval modalities. The paper ends with concluding remarks and open questions.

2 Preliminaries

2.1 Syntax and semantics of propositional neighborhood logics

We will distinguish three variants of propositional neighborhood logics. The language of *full Propositional Neighborhood Logic* ($\mathrm{PNL}^{\pi+}$) consists of a set $\mathcal{AP}$ of atomic propositions (or propositional variables), the propositional connectives $\neg, \vee$, the modal constant π, and the modal operators $\Diamond_r$ and $\Diamond_l$. The other propositional connectives, as well as the logical constants $\top$ (*true*) and $\bot$ (*false*) and the dual modal operators $\Box_r$ and $\Box_l$, are defined as usual. The *formulas* of $\mathrm{PNL}^{\pi+}$, typically denoted by $\varphi, \psi, \ldots$, are recursively defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \pi \mid \Diamond_r\varphi \mid \Diamond_l\varphi.$$

Removing the modal constant π from $\mathrm{PNL}^{\pi+}$ yields the language of *Non-strict Propositional Neighborhood Logic* (PNL^{+}), while the language of *Strict Propositional Neighborhood Logic* (PNL^{-}) is obtained from that of PNL^{+} by replacing the modalities $\Diamond_r$ and $\Diamond_l$ with the modalities $\langle A\rangle$ and $\langle\overline{A}\rangle$ (with dual modalities $[A]$ and $[\overline{A}]$), respectively[5]. We will use PNL to refer collectively to $\mathrm{PNL}^{\pi+}$, PNL^{+}, and PNL^{-}.

Propositional neighborhood logics are interpreted in interval structures on linear orders, which are defined as follows. Let $\mathbb{D} = \langle D, <\rangle$ be a linearly ordered set. An *interval* in $\mathbb{D}$ is an ordered pair $[a, b]$, where $a, b \in D$ and $a \leq b$. An interval $[a, b]$ is a *strict interval* if $a < b$, while it is a *point interval* if $a = b$. We denote the set of all (resp., strict) intervals in $\mathbb{D}$ by $\mathbb{I}(\mathbb{D})^{+}$ (resp., $\mathbb{I}(\mathbb{D})^{-}$). The semantics of $\mathrm{PNL}^{\pi+}/\mathrm{PNL}^{+}$ is given in terms of *non-strict interval models* $\langle\mathbb{I}(\mathbb{D})^{+}, V\rangle$, while that of PNL^{-} is given in terms of *strict interval models* $\langle\mathbb{I}(\mathbb{D})^{-}, V\rangle$. The *valuation function* $V : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})^{+}}$ (resp., $V : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})^{-}}$) assigns to every propositional variable p the set of all (resp., strict) intervals $V(p)$ on which p holds. To explicitly distinguish valuations in non-strict and strict models, we will write V^{+} and V^{-}, respectively; likewise, we will write $\mathbb{I}(\mathbb{D})$ for either of $\mathbb{I}(\mathbb{D})^{+}$ and $\mathbb{I}(\mathbb{D})^{-}$ and will denote non-strict and strict models respectively by $\mathbf{M}^{+}$ and $\mathbf{M}^{-}$, while using $\mathbf{M}$ to denote either type.

The *truth relation* of a formula of PNL at a given interval in a model $\mathbf{M}$ is defined by structural induction on formulas:

- $\mathbf{M}, [a, b] \Vdash p$ iff $[a, b] \in V(p)$, for all $p \in \mathcal{AP}$;
- $\mathbf{M}, [a, b] \Vdash \neg\psi$ iff it is not the case that $\mathbf{M}, [a, b] \Vdash \psi$;
- $\mathbf{M}, [a, b] \Vdash \varphi \vee \psi$ iff $\mathbf{M}, [a, b] \Vdash \varphi$ or $\mathbf{M}, [a, b] \Vdash \psi$;
- $\mathbf{M}, [a, b] \Vdash \Diamond_r\psi$ (resp., $\langle A\rangle\psi$) iff there exists c such that $c \geq b$ (resp., $c > b$) and $\mathbf{M}, [b, c] \Vdash \psi$;
- $\mathbf{M}, [a, b] \Vdash \Diamond_l\psi$ (resp., $\langle\overline{A}\rangle\psi$) iff there exists c such that $c \leq a$ (resp., $c < a$) and $\mathbf{M}, [c, a] \Vdash \psi$;
- $\mathbf{M}^{+}, [a, b] \Vdash \pi$ iff $a = b$.

A PNL-formula is *satisfiable* if it is true on some interval in some interval model for the respective language, and it is *valid* if it is true on every interval in every interval model. It is worth noting that valuation sets represent *binary relations* and thus validity of a PNL-formula is *not* a *monadic second-order* property, but a *dyadic* one.

As shown in [50], PNL can express meaningful temporal properties, e.g., constraints on the structure of the underlying linear ordering. In particular, in $\mathrm{PNL}^{\pi+}$ and PNL^{-} one can express the *difference* operator and thus simulate *nominals*.

2.2 The two-variable fragment of first-order logic

Let us denote by FO^2 (resp., $\mathrm{FO}^2[=]$) the fragment of a generic first-order language (resp., first-order language with equality) whose formulas contain only two fixed distinct variables.

[5] We adopt different notation for the modalities of $\mathrm{PNL}^{\pi+}/\mathrm{PNL}^{+}$ and PNL^{-} to reflect their historical links and to make it easier to distinguish between the non-strict and strict semantics from the syntax.

We denote formulas from these languages by $\alpha, \beta, \ldots$. For example, the formula $\forall x(P(x) \rightarrow \forall y \exists x Q(x, y))$ belongs to FO^2, while the formula $\forall x(P(x) \rightarrow \forall y \exists z(Q(z, y) \wedge Q(z, x)))$ does not. We focus our attention on the language $\mathrm{FO}^2[<]$ over a purely relational vocabulary $\{=, <, P, Q, \ldots\}$ including equality and a distinguished binary relation $<$ interpreted as a linear ordering. Since atoms in the two-variable fragment can involve at most two distinct variables, we may further assume without loss of generality that the arity of every relation is exactly 2.
Let x and y be the two variables of the language. Formulas of $\mathrm{FO}^2[<]$ can be defined recursively as follows:

$$\begin{aligned} \alpha &::= A_0 \mid A_1 \mid \neg\alpha \mid \alpha \vee \beta \mid \exists x\alpha \mid \exists y\alpha \\ A_0 &::= x = x \mid x = y \mid y = x \mid y = y \mid x < y \mid y < x \\ A_1 &::= P(x,x) \mid P(x,y) \mid P(y,x) \mid P(y,y), \end{aligned}$$

where A_1 deals with (uninterpreted) binary predicates. For technical convenience, we assume that both variables x and y occur as (possibly vacuous) free variables in every formula $\alpha \in \mathrm{FO}^2[<]$, that is, $\alpha = \alpha(x, y)$.
Formulas of $\mathrm{FO}^2[<]$ are interpreted in *relational models* of the form $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, where $\mathbb{D} = \langle D, < \rangle$ is a linear ordering and $V_{\mathcal{A}}$ is a *valuation function* that assigns to every *binary relation* P a subset of $D \times D$. When we evaluate a formula $\alpha(x, y)$ on a pair of elements a, b, we write $\alpha(a, b)$ for $\alpha[x := a, y := b]$.
The satisfiability problem for FO^2 without equality was proved decidable by Scott [105] by using a satisfiability preserving reduction of any FO^2-formula to a formula of the form $\forall x \forall y \psi_0 \wedge \bigwedge_{i=1}^{m} \forall x \exists y \psi_i$, which belongs to the Gödel's prefix-defined decidable class of first-order formulas [8]. Later, Mortimer extended this result by including equality in the language [83]. More recently, Grädel, Kolaitis, and Vardi improved Mortimer's result by lowering the complexity bound [55]. Finally, by building on techniques from [55] and performing an in-depth analysis of the basic 1-types and 2-types in $\mathrm{FO}^2[<]$-models, Otto proved the decidability of $\mathrm{FO}^2[<]$ over the class of all linear orderings, as well as over some natural subclasses of it [92].

Theorem 45 ([92]). *The satisfiability problem for* $\mathrm{FO}^2[<]$ *is decidable in NEXPTIME for each of the classes of structures where* $<$ *is interpreted as:*

(i) any linear ordering,
(ii) any well-ordering,
(iii) any finite linear ordering,
(iv) the linear ordering on natural numbers.

2.3 Comparing the expressive power of interval logics

There are different ways to compare the expressive power of different modal languages and logics. For instance, they can be compared with respect to frame validity, that is, with respect to the properties of frames that they can express. (Such a comparison for PNL can be found in [50].) Here we compare the considered logics with respect to expressing properties of a given interval in a model. We distinguish three different cases: the case in which we compare two interval logics on the same class of models, e.g., $\mathrm{PNL}^{\pi+}$ and PNL^{+}, the case in which we compare strict and non-strict interval logics, e.g., PNL^{-} and $\mathrm{PNL}^{\pi+}$, and the case in which we compare an interval logic with a first-order logic, e.g., PNL$^{\pi+}$ and FO$^2[<]$.

Given two interval logics L and L' interpreted in the same class of models $\mathcal{C}$, we say that L' is *at least as expressive as* L (with respect to $\mathcal{C}$), denoted by L $\preceq_{\mathcal{C}}$ L$'$ ($\mathcal{C}$ is omitted if clear from the context), if there exists an effective translation τ from L to L' (inductively defined on the structure of formulas) such that for every model $\mathbf{M}$ in $\mathcal{C}$, any interval $[a,b]$ in $\mathbf{M}$, and any formula φ of L, we have $\mathbf{M},[a,b] \Vdash \varphi$ iff $\mathbf{M},[a,b] \Vdash \tau(\varphi)$. Furthermore, we say that L is *as expressive as* L', denoted by L $\equiv_{\mathcal{C}}$ L$'$, if both L $\preceq_{\mathcal{C}}$ L$'$ and L$'$ $\preceq_{\mathcal{C}}$ L, while we say that L is *strictly more expressive than* L', denoted by L$'$ $\prec_{\mathcal{C}}$ L, if L$'$ $\preceq_{\mathcal{C}}$ L and L $\not\preceq_{\mathcal{C}}$ L$'$.

When comparing an interval logic L$^-$ interpreted in *strict* interval models with an interval logic L$^+$ interpreted in *non-strict* ones, we need to slightly revise the above definitions. Given a strict interval model $\mathbf{M}^- = \langle \mathbb{I}(\mathbb{D})^-, V^- \rangle$, we say that a non-strict interval model $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{D})^+, V^+ \rangle$ is *a non-strict extension* of $\mathbf{M}^-$ (and that $\mathbf{M}^-$ is *the strict restriction* of $\mathbf{M}^+$) if V^- and V^+ agree on the valuation of strict intervals, that is, if for every strict interval $[a,b] \in \mathbb{I}(\mathbb{D})^-$ and propositional variable $p \in \mathcal{AP}$, $[a,b] \in V^-(p)$ if and only if $[a,b] \in V^+(p)$. We say that L$^+$ is *at least as expressive as* L$^-$, and we denote it by L$^-$ $\preceq_I$ L$^+$, if there exists an effective translation τ from L$^-$ to L$^+$ such that for any strict interval model $\mathbf{M}^-$, any interval $[a,b]$ in $\mathbf{M}^-$, and any formula φ of L$^-$, $\mathbf{M}^-,[a,b] \Vdash \varphi$ iff $\mathbf{M}^+,[a,b] \Vdash \tau(\varphi)$ for every non-strict extension $\mathbf{M}^+$ of $\mathbf{M}^-$. Conversely, we say that L$^-$ is *at least as expressive as* L$^+$, and we denote it by L$^+$ $\preceq_I$ L$^-$, if there exists an effective translation τ' from L$^+$ to L$^-$ such that for any non-strict interval model $\mathbf{M}^+$, any strict interval $[a,b]$ in $\mathbf{M}^+$, and any formula φ of L$^+$, $\mathbf{M}^+,[a,b] \Vdash \varphi$ iff $\mathbf{M}^-,[a,b] \Vdash \tau'(\varphi)$, where $\mathbf{M}^-$ is the strict restriction of $\mathbf{M}^+$. L$^-$ $\equiv_I$ L$^+$, L$^-$ $\prec_I$ L$^+$, and L$^+$ $\prec_I$ L$^-$ are defined in the usual way.

Finally, we compare interval logics with first-order logics interpreted in relational models. In this case, the above criteria are no longer adequate, since we need to compare logics which are interpreted in different types of models (interval models and relational models). We deal with this complication by following the approach outlined by Venema in [110]. First, we define suitable model transformations (from interval models to relational models and vice versa); then, we compare the expressiveness of interval and first-order logics modulo these transformations. To define the mapping from interval models to relational models, we associate a binary relation P with every propositional variable $p \in \mathcal{AP}$ of the considered interval logic [110].

Definition 35. *Given an interval model $\boldsymbol{M} = \langle \mathbb{I}(\mathbb{D}), V_{\boldsymbol{M}} \rangle$, the corresponding relational model $\eta(\boldsymbol{M})$ is a pair $\langle \mathbb{D}, V_{\eta(\boldsymbol{M})} \rangle$, where for all $p \in \mathcal{AP}$, $V_{\eta(\boldsymbol{M})}(P) = \{(a,b) \in D \times D : [a,b] \in V_{\boldsymbol{M}}(p)\}$.*

Note that the relational models above can be viewed as 'point' models for modal logics on $\mathbb{D}^2$ and the above transformation as a mapping of propositional variables of the interval logic, interpreted in $\mathbb{I}(\mathbb{D})$, into propositional variables of the target logic, interpreted in $\mathbb{D}^2$ [106,109].

To define the mapping from relational models to interval ones, we have to solve a technical problem: the truth of formulas in interval models is evaluated only on ordered pairs $[a,b]$, with $a \leq b$, while in relational models there is not such a constraint. To deal with this problem, we associate two propositional variables $p^{\leq}$ and $p^{\geq}$ of the interval logic with every binary relation P.

Definition 36. *Given a relational model $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, the corresponding non-strict interval model $\zeta(\mathcal{A})$ is a pair $\langle \mathbb{I}(\mathbb{D})^+, V_{\zeta(\mathcal{A})} \rangle$ such that for any binary relation P and any interval $[a, b]$, $[a, b] \in V_{\zeta(\mathcal{A})}(p^{\leq})$ iff $(a, b) \in V_{\mathcal{A}}(P)$ and $[a, b] \in V_{\zeta(\mathcal{A})}(p^{\geq})$ iff $(b, a) \in V_{\mathcal{A}}(P)$.*

Given an interval logic L_I and a first-order logic L_{FO}, we say that L_{FO} is *at least as expressive as* L_I, denoted by $\mathrm{L}_I \preceq_R \mathrm{L}_{FO}$, if there exists an effective translation τ from L_I to L_{FO} such that for any interval model $\mathbf{M}$, any interval $[a, b]$, and any formula φ of L_I, $\mathbf{M}, [a, b] \Vdash \varphi$ iff $\eta(\mathbf{M}) \models \tau(\varphi)(a, b)$. Conversely, we say that L_I is *at least as expressive as* L_{FO}, denote by $\mathrm{L}_{FO} \preceq_R \mathrm{L}_I$, if there exists an effective translation τ' from L_{FO} to L_I such that for any relational model $\mathcal{A}$, any pair (a, b) of elements, and any formula φ of L_{FO}, $\mathcal{A} \models \varphi(a, b)$ iff $\zeta(\mathcal{A}), [a, b] \Vdash \tau'(\varphi)$ if $a \leq b$ or $\zeta(\mathcal{A}), [b, a] \Vdash \tau'(\varphi)$ otherwise. We say that L_I is *as expressive as* L_{FO}, denoted by $\mathrm{L}_I \equiv_R \mathrm{L}_{FO}$, if $\mathrm{L}_I \preceq_R \mathrm{L}_{FO}$ and $\mathrm{L}_{FO} \preceq_R \mathrm{L}_I$. $\mathrm{L}_I \prec_R \mathrm{L}_{FO}$ and $\mathrm{L}_{FO} \prec_R \mathrm{L}_I$ are defined in the usual way.

3 Comparing the expressiveness of $\mathbf{PNL}^{\pi+}$, $\mathbf{PNL}^{+}$, and $\mathbf{PNL}^{-}$

In this section we compare the relative expressive power of $\mathrm{PNL}^{\pi+}$, PNL^{+}, and PNL^{-}. We will prove that both PNL^{-} and PNL^{+} are strictly less expressive than $\mathrm{PNL}^{\pi+}$, while neither $\mathrm{PNL}^{+} \preceq_I \mathrm{PNL}^{-}$ nor $\mathrm{PNL}^{-} \preceq_I \mathrm{PNL}^{+}$.

In order to compare the expressive power of $\mathrm{PNL}^{\pi+}$ and PNL^{+} we use bisimulation games [54]. More precisely, we apply a simple game-theoretic argument to show two given models that can be distinguished by a $\mathrm{PNL}^{\pi+}$ formula, but not by a PNL^{+} formula. To this end, we define the notion of *k-round PNL^{+}-bisimulation game* to be played by two players, Player I and Player II, on a pair of PNL^{+} models $(\mathbf{M_0}^+, \mathbf{M_1}^+)$, with $\mathbf{M_0}^+ = \langle \mathbb{I}(\mathbb{D}_0)^+, V_0 \rangle$ and $\mathbf{M_1}^+ = \langle \mathbb{I}(\mathbb{D}_1)^+, V_1 \rangle$. The game starts from a given *initial configuration*, where a *configuration* is a pair of intervals $([a_0, b_0], [a_1, b_1])$, with $[a_0, b_0] \in \mathbb{I}(\mathbb{D}_0)^+$ and $[a_1, b_1] \in \mathbb{I}(\mathbb{D}_1)^+$. A configuration $([a_0, b_0], [a_1, b_1])$ is *matching* if $[a_0, b_0]$ and $[a_1, b_1])$ satisfy the same atomic propositions in their respective models.

At every round, given a current configuration $([a_0, b_0], [a_1, b_1])$, Player I can play one of the following two moves:

$\Diamond_r$**-move:** choose $\mathbf{M_i}^+$, where $i \in \{0, 1\}$, and an interval $[b_i, c_i]$;
$\Diamond_l$**-move:** choose $\mathbf{M_i}^+$, where $i \in \{0, 1\}$, and an interval $[c_i, a_i]$.

In the first case, Player II must reply by choosing an interval $[b_{1-i}, c_{1-i}]$ in $\mathbf{M_{1-i}}^+$, which leads to the new configuration $([b_0, c_0], [b_1, c_1])$; likewise, in the second case, Player II must choose an interval $[c_{1-i}, a_{1-i}]$ in $\mathbf{M_{1-i}}^+$ which leads to the new configuration $([c_0, a_0], [c_1, a_1])$. If after any given round the current configuration is not matching, Player I wins the game; otherwise, after k rounds Player II wins the game.

Intuitively, Player II has a *winning strategy* in the k-round PNL^{+}-bisimulation game on the models $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ with a given initial configuration if she can win regardless of the moves played by Player I; otherwise, Player I has a winning strategy. A formal definition of winning strategy can be found in [54]. The following key property of the k-round PNL^{+}-bisimulation game can now be proved routinely, following similar results about bisimulation games in modal logic [54].

Proposition 9. *Let $\mathcal{P}$ be a finite set of propositional variables. For all $k \geq 0$, Player II has a winning strategy in the k-round PNL^{+}-bisimulation game on $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ with initial configuration $([a_0, b_0], [a_1, b_1])$ iff $[a_0, b_0]$ and $[a_1, b_1]$ satisfy the same PNL^{+}-formulas over $\mathcal{P}$ with modal depth at most k.*

To begin with, we will use Proposition 9 to prove that the interval constant π cannot be expressed in PNL$^+$. For that, it suffices to construct two models $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ that can be distinguished with a PNL$^{\pi+}$ formula (which makes an essential use of π), but not by a PNL$^+$ formula. The latter claim is proved by showing that, for all k, Player II has a winning strategy in the k-round PNL$^+$-bisimulation game on $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$.

Theorem 46. *The interval constant π cannot be defined in* PNL$^+$.

Proof. Let $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{Z})^+, V\rangle$, where V is such that p holds everywhere, be a non-strict model. Consider the k-round PNL$^+$-bisimulation game on $(\mathbf{M}^+, \mathbf{M}^+)$ with initial configuration $([0,1],[1,1])$. The intervals $[0,1]$ and $[1,1]$ are easily distinguished in PNL$^{\pi+}$, since π holds in $[1,1]$ but not in $[0,1]$. We show that this pair of intervals cannot be distinguished in PNL$^+$ by providing a simple winning strategy for Player II in the k-round PNL$^+$-bisimulation game on $(\mathbf{M}^+, \mathbf{M}^+)$ with initial configuration $([0,1],[1,1])$. If Player I plays a $\Diamond_r$-move on a given structure, then Player II chooses arbitrarily a right neighbor of the current interval on the other structure. Likewise, if Player I plays a $\Diamond_l$-move on a given structure, then Player II chooses arbitrarily a left-neighbor of the current interval on the other structure. Since the valuation V is such that p holds everywhere, in any case the new configuration is matching. $\square$

Theorem 47. PNL$^-$ $\prec_I$ PNL$^{\pi+}$.

Proof. We prove the claim by showing that PNL$^-$ $\preceq_I$ PNL$^{\pi+}$ and PNL$^{\pi+}$ $\not\preceq_I$ PNL$^-$. To prove the former, we provide a translation τ from PNL$^-$ to PNL$^{\pi+}$. Consider the mapping τ_0 defined as follows:

$$\begin{array}{rlrl} \tau_0(p) = p & & \tau_0(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau_0(\varphi)) \\ \tau_0(\neg\varphi) = \neg\tau_0(\varphi) & & \tau_0(\langle \overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau_0(\varphi)) \\ \tau_0(\varphi_1 \vee \varphi_2) = \tau_0(\varphi_1) \vee \tau_0(\varphi_2) & & \end{array}$$

For every PNL$^-$-formula φ, let $\tau(\varphi) = \neg\pi \wedge \tau_0(\varphi)$. Given a strict model $\mathbf{M}^- = \langle \mathbb{I}(\mathbb{D})^-, V^-\rangle$, let $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{D})^+, V^+\rangle$ be a non-strict extension of $\mathbf{M}^-$. It is immediate to show that for any interval $[a,b]$ in $\mathbf{M}^-$ and any PNL$^-$-formula φ, $\mathbf{M}^-, [a,b] \Vdash \varphi$ if and only if $\mathbf{M}^+, [a,b] \Vdash \tau(\varphi)$. The proof is an easy induction on the structure of φ. This proves that PNL$^-$ $\preceq_I$ PNL$^{\pi+}$. To prove that PNL$^{\pi+}$ $\not\preceq_I$ PNL$^-$, suppose by contradiction that there exists a translation τ' from PNL$^{\pi+}$ to PNL$^-$ such that, for any non-strict model $\mathbf{M}^+$, any strict interval $[a,b]$, and any formula φ of PNL$^{\pi+}$, $\mathbf{M}^+, [a,b] \Vdash \varphi$ iff $\mathbf{M}^-, [a,b] \Vdash \tau'(\varphi)$, where $\mathbf{M}^-$ is the strict restriction of $\mathbf{M}^+$. Consider the non-strict models $\mathbf{M}_0^+ = \langle \mathbb{I}(\mathbb{Z})^+, V_0\rangle$ and $\mathbf{M}_1^+ = \langle \mathbb{I}(\mathbb{Z})^+, V_1\rangle$, where $V_0(p) = \{[a,b] \in \mathbb{I}(\mathbb{Z})^+ : a \leq b\}$ and $V_1(p) = \{[a,b] \in \mathbb{I}(\mathbb{Z})^+ : a < b\}$. It is immediate to see that $\mathbf{M}_0^+, [0,1] \Vdash \Box_r p$, while $\mathbf{M}_1^+, [0,1] \not\Vdash \Box_r p$. Let $\mathbf{M}^- = \langle \mathbb{I}(\mathbb{Z})^-, V^-\rangle$ be a strict interval model such that p holds everywhere in $\mathbb{I}(\mathbb{Z})^-$. We have that $\mathbf{M}^-$ is the strict restriction of both $\mathbf{M}_0^+$ and $\mathbf{M}_1^+$. Hence, we conclude that $\mathbf{M}^-, [0,1] \Vdash \tau'(\Box_r p)$ and $\mathbf{M}^-, [0,1] \not\Vdash \tau'(\Box_r p)$, which is a contradiction. $\square$

Theorem 48. *The expressive powers of* PNL$^+$ *and* PNL$^-$ *are incomparable, namely,* PNL$^-$ $\not\preceq_I$ PNL$^+$ *and* PNL$^+$ $\not\preceq_I$ PNL$^-$.

Proof. We first prove that PNL$^-$ $\not\preceq_I$ PNL$^+$. Let $\mathbf{M_0}^+ = \langle \mathbb{I}(\mathbb{Z})^+, V_0\rangle$ and $\mathbf{M_1}^+ = \langle \mathbb{I}(\mathbb{Z} \setminus \{2\})^+, V_1\rangle$, where V_0 is such that $V_0(p) = \{[1,1],[1,2],[2,2]\}$ and V_1 is such that $V_1(p) = \{[1,1]\}$, be two PNL$^+$-models. For any $k \geq 0$, consider the k-round PNL$^+$-bisimulation

game between $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$, with initial configuration $([0,1],[0,1])$. Player II has the following winning strategy: at any round, if Player I chooses an interval $[a,b] \in \mathbb{I}(\mathbb{Z} \setminus \{2\})^+$ in one of the models, then Player II chooses the same interval on the other model, while if Player I chooses an interval $[a,2]$, with $a < 2$ (resp., $[2,2],[2,b]$, with $b > 2$) in $\mathbf{M_0}^+$, then Player II chooses the interval $[a,1]$ (resp., $[1,1],[1,b]$) in $\mathbf{M_1}^+$. On the other hand, the strict restrictions $\mathbf{M_0}^-$ of $\mathbf{M_0}^+$ and $\mathbf{M_1}^-$ of $\mathbf{M_1}^+$ can be easily distinguished by PNL$^-$: we have that $\mathbf{M_0}^-, [0,1] \Vdash \langle A\rangle p$, while $\mathbf{M_1}^-, [0,1] \nVdash \langle A\rangle p$. Since $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ satisfy the same formulas on the interval $[0,1]$, there cannot exist a translation τ' from PNL$^-$ to PNL$^+$ such that $\mathbf{M_0}^+, [0,1] \Vdash \tau'(\langle A\rangle p)$ and $\mathbf{M_1}^+, [0,1] \nVdash \tau'(\langle A\rangle p)$.
As for PNL$^+ \not\preceq_I$ PNL$^-$, we can use the very same proof we gave to show that PNL$^{\pi+} \not\preceq_I$ PNL$^-$; it suffices to notice that $\Box_r p$ is a PNL$^+$ formula. □

4 Decidability of PNL

In this section, we prove the decidability of PNL$^{\pi+}$, and consequently that of its fragments PNL$^+$ and PNL$^-$, by embedding it into the two-variable fragment of first-order logic FO$^2[<]$ as follows. Let $\mathcal{AP}$ be the set of propositional variables in PNL$^{\pi+}$. The signature for FO$^2[<]$ includes a binary relational symbol P for every $p \in \mathcal{AP}$. The translation function $ST_{x,y}$ is defined as follows:

$$ST_{x,y}(\varphi) = x \leq y \wedge ST'_{x,y}(\varphi),$$

where x, y are two first-order variables and

$$\begin{array}{ll} ST'_{x,y}(p) = P(x,y) & ST'_{x,y}(\varphi \vee \psi) = ST'_{x,y}(\varphi) \vee ST'_{x,y}(\psi) \\ ST'_{x,y}(\pi) = (x = y) & ST'_{x,y}(\Diamond_r \varphi) = \exists x(y \leq x \wedge ST'_{y,x}(\varphi)) \\ ST'_{x,y}(\neg\varphi) = \neg ST'_{x,y}(\varphi) & ST'_{x,y}(\Diamond_l \varphi) = \exists y(y \leq x \wedge ST'_{y,x}(\varphi)) \end{array}$$

Two variables are thus sufficient to translate PNL$^{\pi+}$ into FO$^2[<]$. As we will show later, this is not the case with any proper extension of PNL$^{\pi+}$ in HS or CDT. The next theorem proves that FO$^2[<]$ is at least as expressive as PNL$^{\pi+}$. (Recall that η is the model transformation defined in Section 2.)

Theorem 49. *For any* PNL$^{\pi+}$*-formula* φ*, any non-strict interval model* $\mathbf{M}^+ = \langle \mathbb{I}(\mathbb{D})^+, V\rangle$*, and any interval* $[a,b]$ *in* $\mathbf{M}^+$*:*

$$\mathbf{M}^+, [a,b] \Vdash \varphi \text{ iff } \eta(\mathbf{M}^+) \models ST_{x,y}(\varphi)[x := a, y := b].$$

Proof. The proof is by structural induction on φ. The base case and the cases of the Boolean connectives are straightforward, and thus omitted. Let $\varphi = \Diamond_r \psi$. From $\mathbf{M}^+, [a,b] \Vdash \varphi$, it follows that there exists an element $c \geq b$ such that $\mathbf{M}^+, [b,c] \Vdash \psi$. By inductive hypothesis, we have that $\eta(\mathbf{M}^+) \models ST_{y,x}(\psi)[y := b, x := c]$. By definition of $ST_{y,x}(\psi)$, this is equivalent to $\eta(\mathbf{M}^+) \models y \leq x \wedge ST'_{y,x}(\psi)[y := b, x := c]$. This implies that $\eta(\mathbf{M}^+) \models \exists x(y \leq x \wedge ST'_{y,x}(\psi))[y := b]$. Since $[a,b]$ in $\mathbf{M}^+$, we have $a \leq b$, hence $\eta(\mathbf{M}^+) \models ST_{x,y}(\Diamond_r \psi)[x := a, y := b]$. The converse direction can be proved in a similar way. The case $\varphi = \Diamond_l \psi$ is completely analogous and thus omitted. □

Corollary 8. *A* PNL$^{\pi+}$*-formula* φ *is satisfiable in a class of non-strict interval structures built over a class of linear orderings* $\mathcal{C}$ *iff* $ST_{x,y}(\varphi)$ *is satisfiable in the class of all* FO$^2[<]$*-models expanding linear orderings from* $\mathcal{C}$*.*

Since the above translation is polynomial in the size of the input formula, the complexity of the satisfiability of $\mathrm{PNL}^{\pi+}$ follows from Theorem 45.

Corollary 9. *The satisfiability problem for* $\mathrm{PNL}^{\pi+}$ *is decidable in NEXPTIME for each of the classes of non-strict interval structures built over:*

(i) any linear ordering,
(ii) any well-ordering,
(iii) any finite linear ordering,
(iv) the linear ordering on natural numbers.

Since $\mathrm{PNL}^{+} \prec \mathrm{PNL}^{\pi+}$ and $\mathrm{PNL}^{-} \prec_I \mathrm{PNL}^{\pi+}$, both PNL^{+} and PNL^{-} are decidable in NEXPTIME (at least) on the same classes of orderings as $\mathrm{PNL}^{\pi+}$. Moreover, a translation from PNL^{+} to $\mathrm{FO}^2[<]$ can be obtained from that for $\mathrm{PNL}^{\pi+}$ by simply removing the rule for π, while a translation from PNL^{-} to $\mathrm{FO}^2[<]$ can be obtained from that for $\mathrm{PNL}^{\pi+}$ by removing the rule for π, substituting $<$ for $\leq$, and replacing $\Diamond_r$ with $\langle A \rangle$ and $\Diamond_l$ with $\langle \overline{A} \rangle$. The NEXPTIME-hardness of the satisfiability problem for $\mathrm{PNL}^{\pi+}$, PNL^{+}, and PNL^{-} can be proved by exploiting the reduction from the exponential tiling problem defined by Bresolin et al. for the future fragment of PNL [21]. Together with Corollary 9, such a reduction allows us to prove the following theorem.

Theorem 50. *The satisfiability problem for* PNL^{-}, PNL^{+}, *and* $\mathrm{PNL}^{\pi+}$ *interpreted in the class of all linear orderings (resp., all well-orderings, all finite linear orderings, and the linear ordering on natural numbers) is NEXPTIME-complete.*

This result can be extended to the satisfiability problem for $\mathrm{PNL}^{\pi+}$ in any class of linear orderings definable in $\mathrm{FO}^2[<]$ within any of the above, e.g., the class of all bounded or unbounded (above, below) linear orderings, well-orderings, etc. Moreover, the case of the linear ordering on integer numbers has been positively solved by Bresolin et al. in [20]. On the other hand, the decidability of the satisfiability problem for $\mathrm{PNL}^{\pi+}$ on any of the classes of all discrete, dense, or Dedekind complete linear orderings is still open.

5 Expressive completeness of $\mathrm{PNL}^{\pi+}$ for $\mathrm{FO}^2[<]$

In this section we define a truth preserving translation of $\mathrm{FO}^2[<]$ into $\mathrm{PNL}^{\pi+}$, thus showing that $\mathrm{PNL}^{\pi+}$ is at least as expressive as $\mathrm{FO}^2[<]$. Combining this result with the standard translation of $\mathrm{PNL}^{\pi+}$ into $\mathrm{FO}^2[<]$ presented in the previous section, we conclude that $\mathrm{PNL}^{\pi+}$ is as expressive as $\mathrm{FO}^2[<]$. A similar result was obtained by Venema in [110], viz., the expressive completeness of CDT with respect to the fragment $\mathrm{FO}^3_{x,y}[<]$ of first-order logic interpreted in linear orderings, whose language contains only three, possibly reused variables and at most two of them, x and y, can be free in a formula. Both results can be viewed as interval-based counterparts of Kamp's expressive completeness theorem for the propositional point-based linear time temporal logic LTL with respect to the monadic first-order logic over Dedekind complete linear orderings [70].

The translation τ from $\mathrm{FO}^2[<]$ to $\mathrm{PNL}^{\pi+}$ is given in the following table:

Basic formulas	Non-basic formulas
$\tau[x,y](x=x) = \tau[x,y](y=y) = \top$	$\tau[x,y](\neg\alpha) = \neg\tau[x,y](\alpha)$
$\tau[x,y](x=y) = \tau[x,y](y=x) = \pi$	$\tau[x,y](\alpha \vee \beta) = \tau[x,y](\alpha) \vee \tau[x,y](\beta)$
$\tau[x,y](y<x) = \bot$	$\tau[x,y](\exists x\beta) =$
$\tau[x,y](x<y) = \neg\pi$	$\Diamond_r(\tau[y,x](\beta)) \vee \Box_r\Diamond_l(\tau[x,y](\beta))$
$\tau[x,y](P(x,x)) = \Diamond_l(\pi \wedge p^{\leq} \wedge p^{\geq})$	$\tau[x,y](\exists y\beta) =$
$\tau[x,y](P(y,y)) = \Diamond_r(\pi \wedge p^{\leq} \wedge p^{\geq})$	$\Diamond_l(\tau[y,x](\beta)) \vee \Box_l\Diamond_r(\tau[x,y](\beta))$
$\tau[x,y](P(x,y)) = p^{\leq}$	
$\tau[x,y](P(y,x)) = p^{\geq}$	

As formally stated by Theorem 51 below, every $\mathrm{FO}^2[<]$-formula $\alpha(x,y)$ is mapped into two distinct $\mathrm{PNL}^{\pi+}$-formulas $\tau[x,y](\alpha)$ and $\tau[y,x](\alpha)$. The first one captures precisely those models (if any) of $\alpha(x,y)$ where $x \leq y$, while the second one captures precisely those models (if any) of $\alpha(x,y)$ where $y \leq x$.

Example 1. Consider the formula $\alpha = \exists x \neg \exists y(x<y)$, which constrains the model to be bounded above. Let $\beta = \exists y(x<y)$. We have that

$$\begin{aligned}\tau[x,y](\beta) = \Diamond_l(\tau[y,x](x<y)) \vee \Box_l\Diamond_r(\tau[x,y](x<y)) \ = \\ = \Diamond_l\bot \vee \Box_l\Diamond_r\neg\pi \ (\equiv \ \Box_l\Diamond_r\neg\pi)\end{aligned}$$

and that

$$\begin{aligned}\tau[y,x](\beta) = \Diamond_r(\tau[x,y](x<y)) \vee \Box_r\Diamond_l(\tau[y,x](x<y)) \ = \\ = \Diamond_r\neg\pi \vee \Box_r\Diamond_l\bot \ (\equiv \ \Diamond_r\neg\pi).\end{aligned}$$

The resulting translation of α is:

$$\begin{aligned}\tau[x,y](\alpha) &= \Diamond_r(\tau[y,x](\neg\beta)) \vee \Box_r\Diamond_l(\tau[x,y](\neg\beta)) \ = \\ &= \Diamond_r(\neg\tau[y,x](\beta)) \vee \Box_r\Diamond_l(\neg\tau[x,y](\beta)) \ = \\ &= \Diamond_r\neg\Diamond_r\neg\pi \vee \Box_r\Diamond_l\neg\Box_l\Diamond_r\neg\pi \ = \\ &= \Diamond_r\Box_r\pi \vee \Box_r\Diamond_l\Diamond_l\Box_r\pi \\ &\quad (\equiv \Diamond_r\Box_r\pi \vee \Box_r\pi).\end{aligned}$$

which is a $\mathrm{PNL}^{\pi+}$-formula that, likewise, constrains the model to be bounded above.

Given a $\mathrm{FO}^2[<]$-model $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, let $\zeta(\mathcal{A}) = \langle \mathbb{I}(\mathbb{D})^+, V_{\zeta(\mathcal{A})} \rangle$ be the corresponding $\mathrm{PNL}^{\pi+}$-model (see Section 2).

Theorem 51. *For every* $\mathrm{FO}^2[<]$*-formula* $\alpha(x,y)$*, every* $\mathrm{FO}^2[<]$*-model* $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$*, and every pair* $a, b \in D$*, with* $a \leq b$*:*

(i) $\mathcal{A} \models \alpha(a,b)$ *if and only if* $\zeta(\mathcal{A}), [a,b] \Vdash \tau[x,y](\alpha)$*, and*
(ii) $\mathcal{A} \models \alpha(b,a)$ *if and only if* $\zeta(\mathcal{A}), [a,b] \Vdash \tau[y,x](\alpha)$*.*

Proof. The proof is by simultaneous induction on the complexity of α.

- $\alpha = (x = x)$ or $\alpha = (y = y)$. Both α and $\tau[x,y](\alpha) = \top$ are true.

- $\alpha = (x < y)$.
  Claim (i): $\mathcal{A} \models \alpha(a,b)$ iff $a < b$ iff $\zeta(\mathcal{A}), [a,b] \Vdash \neg\pi$.
  Claim (ii): $\mathcal{A} \not\models \alpha(b,a)$, since $a \leq b$, and $\zeta(\mathcal{A}), [a,b] \not\Vdash \tau[y,x](x < y)(= \bot)$. Likewise, for $\alpha = (y < x)$.
- $\alpha = P(x,y)$ or $\alpha = P(y,x)$. Both claims follow from the valuation of $p^{\leq}$ and $p^{\geq}$ (given in Section 2).
- $\alpha = P(x,x)$.
  Claim (i): $\mathcal{A} \models \alpha(a,b)$ iff $\mathcal{A} \models P(a,a)$ iff $\zeta(\mathcal{A}), [a,a] \Vdash \pi \wedge p^{\leq} \wedge p^{\geq}$ iff $\zeta(\mathcal{A}), [a,b] \Vdash \Diamond_l(\pi \wedge p^{\leq} \wedge p^{\geq})$.
  A similar argument can be used to prove claim (ii). Likewise for $\alpha = P(y,y)$.
- The Boolean cases are straightforward.
- $\alpha = \exists x \beta$.
  Claim (i): suppose that $\mathcal{A} \models \alpha(a,b)$. Then, there is $c \in \mathcal{A}$ such that $\mathcal{A} \models \beta(c,b)$. There are two (non-exclusive) cases: $b \leq c$ and $c \leq b$. If $b \leq c$, by the inductive hypothesis, we have that $\zeta(\mathcal{A}), [b,c] \Vdash \tau[y,x](\beta)$ and thus $\zeta(\mathcal{A}), [a,b] \Vdash \Diamond_r(\tau[y,x](\beta))$. Likewise, if $c \leq b$, by the inductive hypothesis, we have that $\zeta(\mathcal{A}), [c,b] \Vdash \tau[x,y](\beta)$ and thus for every d such that $b \leq d$, $\zeta(\mathcal{A}), [b,d] \Vdash \Diamond_l(\tau[x,y](\beta))$, that is, $\zeta(\mathcal{A}), [a,b] \Vdash \Box_r\Diamond_l(\tau[x,y](\beta))$. Hence $\zeta(\mathcal{A}), [a,b] \Vdash \Diamond_r(\tau[y,x](\beta)) \vee \Box_r\Diamond_l(\tau[x,y](\beta))$, that is, $\zeta(\mathcal{A}), [a,b] \Vdash \tau[x,y](\alpha)$. For the converse direction, it suffices to note that the interval $[a,b]$ has at least one right neighbor, viz. $[b,b]$, and thus the above argument can be reversed.
  Claim (ii) can be proved similarly.
- $\alpha = \exists y \beta$. Analogous to the previous case. $\Box$

Corollary 10. *For every formula $\alpha(x,y)$ and every $\mathrm{FO}^2[<]$-model $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, $\mathcal{A} \models \forall x \forall y \alpha(x,y)$ if and only if $\zeta(\mathcal{A}) \Vdash \tau[x,y](\alpha) \wedge \tau[y,x](\alpha)$.*

Definition 37. *We say that a $\mathrm{PNL}^{\pi+}$-model $\boldsymbol{M}$ of the considered language is* synchronized on a pair of variables $(p^{\leq}, p^{\geq})$ *if these variables are equally true at any point interval $[a,a]$ in $\boldsymbol{M}$; $\boldsymbol{M}$ is* synchronized for a $\mathrm{FO}^2[<]$-formula α *if it is synchronized on every pair of variables $(p^{\leq}, p^{\geq})$ corresponding to a predicate p occurring in α; $\boldsymbol{M}$ is* synchronized *if it is synchronized on every pair $(p^{\leq}, p^{\geq})$.*

It is immediate to see that every model $\zeta(\mathcal{A})$, where $\mathcal{A}$ is a $\mathrm{FO}^2[<]$-model, is synchronized. Conversely, every synchronized $\mathrm{PNL}^{\pi+}$-model $\mathbf{M}$ can be represented as $\zeta(\mathcal{A})$ for some model $\mathcal{A}$ for $\mathrm{FO}^2[<]$: the linear ordering of $\mathcal{A}$ is inherited from $\mathbf{M}$ and the interpretation of every binary predicate P is defined in accordance with Theorem 51, that is, for any $a, b \in \mathcal{A}$ we set $P(a,b)$ to be true precisely when $a \leq b$ and $\mathbf{M}, [a,b] \Vdash p^{\leq}$ or $b \leq a$ and $\mathbf{M}, [b,a] \Vdash p^{\geq}$. Due to the synchronization, these two conditions agree when $a = b$. Furthermore, the condition that a $\mathrm{PNL}^{\pi+}$-model $\mathbf{M}$ is synchronized on a pair of variables $p^{\leq}$ and $p^{\geq}$ can be expressed by the validity in $\mathbf{M}$ of the formula $[U](\pi \rightarrow (p^{\leq} \leftrightarrow p^{\geq}))$, where $[U]$ is the *universal modality*, which is definable in $\mathrm{PNL}^{\pi+}$ as follows [50]:

$$[U]\varphi ::= \Box_r\Box_r\Box_l\varphi \wedge \Box_r\Box_l\Box_l\varphi \wedge \Box_l\Box_l\Box_r\varphi \wedge \Box_l\Box_r\Box_r\varphi.$$

Building on this observation, we associate with every $\mathrm{FO}^2[<]$-formula α the formulas

$$\sigma_v(\alpha) = \left(\bigwedge_{p^{\leq}, p^{\geq}} [U](\pi \rightarrow (p^{\leq} \leftrightarrow p^{\geq})) \right) \rightarrow (\tau[x,y](\alpha) \wedge \tau[y,x](\alpha))$$

and

$$\sigma_s(\alpha) = \left(\bigwedge_{p^{\leq}, p^{\geq}} [U](\pi \rightarrow (p^{\leq} \leftrightarrow p^{\geq})) \right) \wedge (\tau[x,y](\alpha) \vee \tau[y,x](\alpha)),$$

where the conjunctions range over all pairs $p^{\leq}, p^{\geq}$ corresponding to predicates occurring in α.

Corollary 11. *For any* $\mathrm{FO}^2[<]$*-formula* α*:*

(i) α *is valid in all* $\mathrm{FO}^2[<]$*-models iff* $\sigma_v(\alpha)$ *is a valid* $\mathrm{PNL}^{\pi+}$*-formula, and*
(ii) α *is satisfiable in some* $\mathrm{FO}^2[<]$*-model iff* $\sigma_s(\alpha)$ *is a satisfiable* $\mathrm{PNL}^{\pi+}$*-formula.*

Note that the proposed translation from $\mathrm{FO}^2[<]$ to $\mathrm{PNL}^{\pi+}$ is exponential in the size of the input formula, due to the clause for the existential quantifier (at the moment, we do not know whether there exists a polynomial translation).

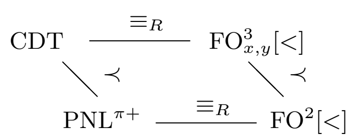


Fig. 7. Expressive completeness results for interval logics.

In Figure 7 we put together the expressive completeness results for CDT and $\mathrm{PNL}^{\pi+}$, using the notation introduced in Section 2. Since $\mathrm{FO}^2[<]$ is a proper fragment of $\mathrm{FO}^3_{x,y}[<]$, from the equivalences between CDT and $\mathrm{FO}^3_{x,y}[<]$ and between $\mathrm{PNL}^{\pi+}$ and $\mathrm{FO}^2[<]$ it immediately follows that CDT is strictly more expressive than $\mathrm{PNL}^{\pi+}$.

6 Comparing $\mathrm{PNL}^{\pi+}$ with other fragments of HS

In this section we compare $\mathrm{PNL}^{\pi+}$ with other fragments of HSand show that $\mathrm{PNL}^{\pi+}$ is essentially the maximal fragment of HSwhich translates to $\mathrm{FO}^2[<]$. More precisely, we consider the interval modalities $\langle B \rangle$, $\langle E \rangle$, $\langle O \rangle$, $\langle D \rangle$, $\langle L \rangle$, and their inverses, corresponding to Allen's relations *begins, ends, overlaps, during, after*, and their inverse relations. The standard translations of these modalities into first-order logic are as follows:

$$ST_{x,y}(\langle B \rangle \varphi) = x \leq y \wedge \exists z(z < y \wedge ST_{x,z}(\varphi))$$
$$ST_{x,y}(\langle E \rangle \varphi) = x \leq y \wedge \exists z(x < z \wedge ST_{z,y}(\varphi))$$
$$ST_{x,y}(\langle O \rangle \varphi) = x \leq y \wedge \exists z(x < z < y \wedge \exists y(y < x \wedge ST_{y,z}(\varphi)))$$
$$ST_{x,y}(\langle D \rangle \varphi) = x \leq y \wedge \exists z(x < z < y \wedge \exists y(x < y \wedge ST_{y,z}(\varphi)))$$
$$ST_{x,y}(\langle L \rangle \varphi) = x \leq y \wedge \exists x(y < x \wedge \exists y ST_{x,y}(\varphi))$$

Note that the standard translation of $\langle L \rangle$ is a two-variable formula, while the standard translations of the other modalities are three-variable formulas. However, $\langle L \rangle$ can be defined in $\mathrm{PNL}^{\pi+}$ as follows: $\langle L \rangle \varphi = \Diamond_r(\neg \pi \wedge \Diamond_r \varphi)$. Likewise, the inverted modality $\langle \overline{L} \rangle$ is definable in $\mathrm{PNL}^{\pi+}$.
We will show that none of the other interval modalities listed above can be defined in $\mathrm{PNL}^{\pi+}$ by using game-theoretic arguments similar to those in the proof of Theorem 46.

To this end, we define the k-*round* PNL$^{\pi+}$-*bisimulation game* played on a pair of PNL$^{\pi+}$ models $(\mathbf{M_0}^+, \mathbf{M_1}^+)$ starting from a given initial configuration as follows. The rules of the game are the same as those of the k-round PNL$^+$-bisimulation game described in Section 3; the only difference is that a configuration $([a_0, b_0], [a_1, b_1])$ is matching if and only if:

(i) $[a_0, b_0]$ and $[a_1, b_1]$ share the same valuation of propositional variables, and
(ii) $a_0 = b_0$ iff $a_1 = b_1$, that is, $\mathbf{M_0}^+, [a_0, b_0] \Vdash \pi$ iff $\mathbf{M_1}^+, [a_1, b_1] \Vdash \pi$.

The following result is analogous to Proposition 9.

Proposition 10. *Let $\mathcal{P}$ be a finite set of propositional variables. For all $k \geq 0$, Player II has a winning strategy in the k-round* PNL$^{\pi+}$*-bisimulation game on $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ with initial configuration $([a_0, b_0], [a_1, b_1])$ iff $[a_0, b_0]$ and $[a_1, b_1]$ satisfy the same formulas of* PNL$^{\pi+}$ *over $\mathcal{P}$ with operator depth at most k.*

We exploit Proposition 10 to prove that none of the interval modalities $\langle B\rangle$, $\langle E\rangle$, $\langle O\rangle$, and $\langle D\rangle$ is expressible in PNL$^{\pi+}$. The proof structure is always the same: for every operator $\langle X\rangle$, we choose two models $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ that can be distinguished with a formula containing $\langle X\rangle$ and we prove that Player II has a winning strategy in the k-rounds PNL$^{\pi+}$-bisimulation game.

Theorem 52. *Neither of $\langle B\rangle$, $\langle E\rangle$, $\langle O\rangle$, and $\langle D\rangle$, or their inverses, can be defined in* PNL$^{\pi+}$.

Proof. We prove the claim for $\langle B\rangle$ and $\langle D\rangle$; the other cases are analogous. Consider the PNL$^{\pi+}$-models $\mathbf{M_0}^+ = \langle \mathbb{I}(\mathbb{Z} \setminus \{1, 2\})^+, V_0\rangle$ and $\mathbf{M_1}^+ = \langle \mathbb{I}(\mathbb{Z})^+, V_1\rangle$, where V_1 is such that p holds for all intervals $[a, b]$ such that $a < b$ and V_0 is the restriction of V_1 to $\mathbb{I}(\mathbb{Z} \setminus \{1, 2\})^+$. Note that $\mathbf{M_1}^+, [0, 3] \Vdash \langle B\rangle p$, while $\mathbf{M_0}^+, [0, 3] \not\Vdash \langle B\rangle p$; likewise for $\langle D\rangle p$. Thus, to prove the claims it suffices to show that Player II has a winning strategy for the k-round PNL$^{\pi+}$-bisimulation game between $\mathbf{M_0}^+$ and $\mathbf{M_1}^+$ with initial configuration $([0, 3], [0, 3])$. In fact, Player II has a *uniform* strategy to play forever that game: at any position, assuming that Player I has not won yet, if he chooses a $\Diamond_r$-move then Player II chooses arbitrarily a right neighbor of the current interval on the other structure, with the only constraint to take a point-interval if and only if Player I has taken a point-interval as well. If Player I chooses a $\Diamond_l$-move, Player II acts likewise. ☐

7 Undecidable extensions of PNL

A natural question now arises: is it possible to extend PNL$^{\pi+}$ with other modal operators (such as those listed in the previous section) without losing decidability? In this section we address and partly answer this question negatively, by considering the extensions of PNL$^{\pi+}$ within HS. First of all, we show that adding to PNL the modal operator $\langle D\rangle$, or its inverse $\langle \overline{D}\rangle$, suffices to cross the undecidability border. The technique used here is based on a non-trivial reduction from the *unbounded tiling problem* for the second octant $\mathcal{O}$ of the integer plane [8]. It works for the class of all linear orders and for a number of interesting subclasses of it. Moreover, it turns out to be quite versatile, being applicable to a variety of extensions of PNL$^{\pi+}$. In summary, we will show that the satisfiability problem for any extension of PNL$^{\pi+}$ containing at least one of the following is undecidable: $\langle D\rangle$, $\langle \overline{D}\rangle$, $\langle D\rangle_{\sqsubset}$, $\langle \overline{D}\rangle_{\sqsubset}$, $\langle B\rangle$ and $\langle \overline{E}\rangle$, $\langle \overline{B}\rangle$ and $\langle E\rangle$, where $\langle D\rangle_{\sqsubset}$ is the modal operator of the *proper subinterval relation* (and $\langle \overline{D}\rangle_{\sqsubset}$ is its inverse), studied in more detail in [12,13], which is defined as follows (in fact, three variables suffice to define $\langle D\rangle_{\sqsubset}$; four variables makes it possible to define it in a more compact way):

$$ST_{x,y}(\langle D\rangle_{\sqsubset}\varphi) = x < y \wedge \exists z \exists w (x \leq z \wedge w \leq y \wedge (x < z \vee w < y) \wedge ST_{z,w}(\varphi))$$

These cases cover a huge majority of all fragments of HScontaining $\text{PNL}^{\pi+}$ (an estimate of 549 out of 576). In the following, we will provide a detailed analysis of the representative case of $\text{PNL}^{\pi+} + \langle D\rangle$; at the end of the section, we will show how to adapt the formulas used in the proof to the remaining cases. Beside the original undecidability result for HS, the present one can be paired with two other undecidability results, namely, that for the BE-fragment [74,52] and that for Compass Logic [76], which can be seen as a generalized propositional logic of intervals.

7.1 Undecidability of $\text{PNL}^{\pi+}$ $\langle D\rangle$

Language and point-intervals. Let us fix an arbitrary finite set of tiles $t_1, \ldots, t_k$ and assume that the set of atomic propositions $\mathcal{AP}$ is finite (but arbitrary) and contains, inter alia, the following propositional variables: u, Id, tile, $\mathsf{right}(t_1)$, …, $\mathsf{right}(t_k)$, $\mathsf{left}(t_1)$, …, $\mathsf{left}(t_k)$, $\mathsf{up}(t_1)$, …, $\mathsf{up}(t_k)$, $\mathsf{down}(t_1)$, …, $\mathsf{down}(t_k)$, T_1,…, T_k, bb, be, eb, and corr. For sake of convenience, we define the PNL^- operator $\langle A\rangle$ in terms of $\Diamond_r$ and π:

$$\langle A\rangle p = \Diamond_r(\neg\pi \wedge p). \tag{1}$$

The inverse operator $\langle\overline{A}\rangle$ can be defined likewise.

Unit-intervals. We set our framework by forcing the existence of a unique infinite chain of so-called *unit-intervals* (for short, u-*intervals*) on the linear order, which covers an initial segment of the model. These u-intervals will be used as cells to arrange the tiling. They will be labeled by the propositional variable u. Formally, we define the formula

$$\mathsf{UnitChain} ::= \mathsf{u} \wedge [\overline{A}][\overline{A}][A]\neg\mathsf{u} \wedge [U](\mathsf{u} \rightarrow (\neg\pi \wedge \langle A\rangle\mathsf{u} \wedge \neg\langle D\rangle\langle A\rangle\mathsf{u})). \tag{2}$$

Lemma 6. *Suppose that* $\mathbf{M}, [a,b] \Vdash \mathsf{UnitChain}$. *Then, there exists an infinite sequence of points* $b_0 < b_1 < \ldots$ *in* $\mathbf{M}$, *such that* $a = b_0, b = b_1$, *for each* i, $\mathbf{M}, [b_i, b_{i+1}] \Vdash u$, *and no other interval* $[c,d]$ *in* $\mathbf{M}$ *satisfies* u, *unless* $c > b_i$ *for every* $i \in \mathbb{N}$.

Proof. Clearly, $\mathbf{M}, [a,b] \Vdash \mathsf{u} \wedge \neg\pi$, so $a \neq b$. The existence of the chain of endpoints of u-intervals $b_1 < b_2 < \ldots$ starting from a, b is easy, because every u-interval has a right neighbor u-interval. We still have to show that no other point either ends or begins a u-interval. Indeed, suppose for some c, d, such that $c \neq b_i$ for every $i = 0, 1, \ldots$, it holds that $\mathbf{M}, [c,d] \Vdash \mathsf{u}$. Because $\mathbf{M}, [a,b] \Vdash [\overline{A}][\overline{A}][A]\neg\mathsf{u}$, we have that $b_0 < c$, hence either $b_i < c < b_{i+1}$ for some i or $c > b_i$ for every $i = 0, 1, \ldots$. In the former case, $\mathbf{M}, [c,c] \Vdash \langle A\rangle\mathsf{u}$, hence $\mathbf{M}, [b_i, b_{i+1}] \Vdash \mathsf{u} \wedge \langle D\rangle\langle A\rangle\mathsf{u}$ which contradicts $\mathbf{M}, [a,b] \Vdash [U](\mathsf{u} \rightarrow \neg\langle D\rangle\langle A\rangle\mathsf{u})$. Finally, notice that the case in which $c = b_i$ and $d = b_{i+q}$, where $q > 1$, contradicts $\mathbf{M}, [a,b] \Vdash [U](\mathsf{u} \rightarrow \neg\langle D\rangle\langle A\rangle\mathsf{u})$, since $\mathbf{M}, [b_{i+1}, b_{i+1}] \Vdash \langle A\rangle\mathsf{u}$. □

Then, to restrict our domain of 'legitimate intervals' to those composed of u-intervals, we impose that every interval of importance begins and ends with a u-interval:

$$[U] \bigwedge_{p \in \mathcal{AP}} (p \rightarrow [\overline{A}]\langle A\rangle\mathsf{u} \wedge [A]\langle\overline{A}\rangle\mathsf{u}). \tag{3}$$

Encoding a tile. Every u-interval will represent either a tile or a special marker, denoted by $*$, indicating the border between two Id-intervals, that will be defined later. Thus, we put:

$$[U](\mathsf{u} \leftrightarrow (* \vee \mathsf{tile}) \wedge (* \leftrightarrow \neg\mathsf{tile})), \tag{4}$$

$$[U](\mathsf{tile} \leftrightarrow (\bigvee_{i=1}^{k} T_i \wedge \bigwedge_{i,j=1, j\neq i}^{k} \neg(T_i \wedge T_j))), \tag{5}$$

where each T_i, for $i = 1 \ldots, k$, is the formula:

$$T_i = \mathsf{right}(t_i) \wedge \mathsf{left}(t_i) \wedge \mathsf{up}(t_i) \wedge \mathsf{down}(t_i). \tag{6}$$

where $\mathsf{right}(t_i), \mathsf{left}(t_i), \mathsf{up}(t_i), \mathsf{down}(t_i)$ represent the colors of the respective sides of the tile t_i.
If a tile is placed on a u-interval $[a, b]$, we call a and b respectively the *beginning point* and the *ending point* of that tile.

Encoding rows of the tiling. Each Id-interval (or just Id) represents a row (level) of the tiling of $\mathcal{O}$. An Id-interval is an interval consisting of a finite sequence of at least two u-subintervals. The first u-subinterval in an Id is a $*$-interval and every following u-subinterval is the encoding of a tile. Moreover, the Id-intervals representing the bottom-up consecutive levels of the tiling of $\mathcal{O}$ are arranged one after another in a chain. So:

$$[U](\mathsf{Id} \rightarrow \neg\mathsf{u} \wedge \langle\overline{A}\rangle\langle A\rangle * \wedge \langle A\rangle\mathsf{Id}). \tag{7}$$

To prevent the existence of interleaving sequences of Id-intervals, we not allow occurrences of $*$-subintervals inside an Id by means of the following formula:

$$[U](\mathsf{Id} \rightarrow \neg\langle D\rangle\langle A\rangle *). \tag{8}$$

The next formula states that the first Id is composed by a single tile:

$$\mathsf{First} = [\overline{A}][\overline{A}][A]\neg\mathsf{u} \wedge \mathsf{Id} \wedge \langle\overline{A}\rangle\langle A\rangle(* \wedge \langle A\rangle\mathsf{tile}). \tag{9}$$

Finally, we put:

$$\mathsf{IdDef} = \mathsf{First} \wedge (2) \wedge (3) \wedge (4) \wedge (5) \wedge (7) \wedge (8). \tag{10}$$

Lemma 7. *Suppose that* $\mathbf{M}, [a, b] \Vdash \mathsf{IdDef}$. *Then, there is a sequence of points* $a = b_1^0 < b_1^1 < \ldots b_1^{k_1} = b_2^0 < b_2^1 < \ldots < b_2^{k_2} = b_3^0 < \ldots$, *such that* $k_1 = 2$ *and for every* j:

1. $\mathbf{M}, [b_j^0, b_j^{k_j}] \Vdash \mathsf{Id}$ *and no other interval* $[c, d]$ *in* $\mathbf{M}$ *is an* Id*-interval, unless possibly for* $c > b_j^{k_j}$ *for every* $j \in \mathbb{N}$.
2. $\mathbf{M}, [b_j^0, b_j^1] \Vdash *$ *and no other interval* $[c, d]$ *in* $\mathbf{M}$ *is a* $*$*-interval, unless possibly for* $c > b_j^{k_j}$ *for every* $j \in \mathbb{N}$.
3. *For every* i *such that* $0 < i < k_j$, $\mathbf{M}, [b_j^i, b_j^{i+1}] \Vdash \mathsf{tile}$, *and no other interval* $[c, d]$ *in* $\mathbf{M}$ *is a* tile*-interval, unless possibly for* $c > b_j^{k_j}$ *for every* $j \in \mathbb{N}$.

Proof. The existence of the infinite sequence of points follows from First and formula (7) which together imply existence of an infinite sequence of consecutive Id-intervals $[b_1^0, b_1^{k_1}], [b_2^0, b_2^{k_2}] \ldots$. Now, let the endpoints of the u-subintervals of $[b_j^0, b_j^{k_j}]$ be $b_j^0 < b_j^1 < \ldots < b_j^{k_j}$. Thus, the first part of claim (1) holds by construction.

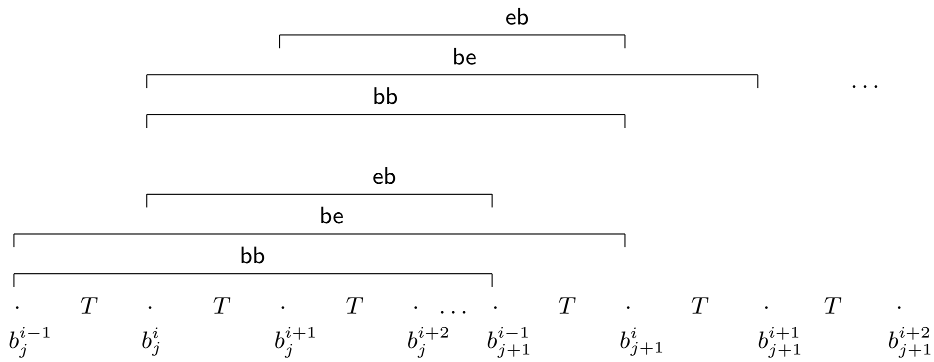


Fig. 8. A representation of bb, be, and eb-intervals.

Now suppose that another interval $[c, d]$ satisfies Id. The left endpoint c cannot be less than a, because an Id-interval begins with a u-interval and, by First, no u-interval begins to the left of a. Assuming that $b_j^0 < c < b_j^{k_j}$ for some j leads to a contradiction with (8), because the beginning of $[c, d]$ is a $*$-interval properly contained in the Id-interval $[b_j^0, b_j^{k_j}]$. Finally, assuming that $c = b_j^0$ for some j leads to a contradiction as well, because $[c, d]$, which is different from $[b_j^0, b_j^{k_j}]$, must be followed immediately by another Id-interval $[d, e]$ and the beginning $*$-subinterval of $[d, e]$ must be strictly inside $[b_j^0, b_j^{k_j}]$ or the $*$-interval $[b_{j+1}^0, b_{j+1}^1]$ must be strictly inside $[c, d]$, either of which is impossible, due to condition (8). Thus, claim (1) is established.
Claims (2) and (3) can be proved in a similar way, using the respective conjuncts in IdDef.
□

Definition 38. *Let* $\mathbf{M}, [a, b] \Vdash \mathsf{IdDef}$ *and* $b_1^0 < b_1^1 < \ldots b_1^{k_1} = b_2^0 < b_2^1 < \ldots < b_2^{k_2} = b_3^0 \ldots$ *be the sequence of points whose existence is guaranteed by Lemma 7. For any* j*, the interval* $[b_j^0, b_j^{k_j}]$ *is the* j-th Id-interval *of the sequence and for any* $i \geq 1$ *the interval* $[b_j^i, b_j^{i+1}]$ *is the* i-th tile *of the* Id*-interval* $[b_j^0, b_j^{k_j}]$.

Corresponding tiles. So far we have that, given a starting interval, the formula IdDef forces the underlying linearly ordered set to be, in the future of the current interval, a sequence of Id's, the first one of which containing exactly one tile. Now, we want to make sure that each tile at a certain *level* in $\mathcal{O}$ (i.e., Id) always has its corresponding tile at the immediate upper level. We will use auxiliary propositional variables in order to guarantee this property, namely: bb, which is to connect the beginning point of a tile to the beginning point of the corresponding tile above; be which is to connect the beginning point of a tile to the ending point of the corresponding tile above; and eb, which is to connect the ending point of a tile to the beginning point of the corresponding tile above. If an interval is labeled with either of these three propositional variables, we call it a *corresponding interval*, abbreviated corr-*interval*. A pictorial representation is given in Figure 8. In the following, we force corr-intervals to respect suitable properties so that all models satisfying them encode correct tiling.
First, we put the propositional variable corr wherever one among bb, be and eb holds:

$$[U]((\mathsf{bb} \vee \mathsf{be} \vee \mathsf{eb}) \leftrightarrow \mathsf{corr}). \tag{11}$$

Then, we prevent any corr-interval to coincide with an Id-interval:

$$[U]\neg(\mathsf{corr} \wedge \mathsf{Id}). \tag{12}$$

In addition, we impose that neither a corr-interval is properly contained in an Id-interval, nor the other way round. This means that a corr-interval must contain a unique separating marker $*$ and that it cannot be followed immediately by $*$:

$$[U](\mathsf{corr} \rightarrow (\neg \mathsf{u} \wedge \langle D\rangle(\langle A\rangle * \vee \langle \overline{A}\rangle *) \wedge \neg\langle D\rangle(\langle A\rangle * \wedge \langle \overline{A}\rangle *) \wedge \neg\langle A\rangle *). \tag{13}$$

We put

$$\mathsf{CorrDef} = (11) \wedge (12) \wedge (13). \tag{14}$$

Lemma 8. *Let* $\mathbf{M}, [a, b] \Vdash \mathsf{IdDef} \wedge \mathsf{CorrDef}$*. Then no* Id*-interval in* $\mathbf{M}$ *coincides with a* corr*-interval, nor is properly contained in a* corr*-interval, nor a* corr*-interval is properly contained in an* Id*-interval, unless it begins with a* $*$*.*

Proof. A corr-interval cannot coincides with an Id-interval because of (12); it cannot properly contain an Id-interval because of (13), and it cannot be properly contained in an Id-interval unless it begins with a $*$, again by (13). □

To guarantee that every tile in every Id corresponds, via bb, be, and eb, to some tile of the next Id and that every tile, but the last one, of every Id corresponds, via bb, be, and eb, to some tile of the previous Id, we take advantage of the following formulas:

$$[U](\mathsf{u} \rightarrow (\neg * \leftrightarrow \langle \overline{A}\rangle\langle A\rangle \mathsf{bb})), \tag{15}$$

$$[U](\mathsf{u} \rightarrow ((\neg\langle A\rangle * \wedge \neg\langle A\rangle(\mathsf{u} \wedge \langle A\rangle *)) \leftrightarrow \langle A\rangle\langle \overline{A}\rangle \mathsf{bb})), \tag{16}$$

$$[U](\mathsf{u} \rightarrow (\neg * \leftrightarrow \langle \overline{A}\rangle\langle A\rangle \mathsf{be})), \tag{17}$$

$$[U](\mathsf{u} \rightarrow ((\neg * \wedge \neg\langle A\rangle *) \leftrightarrow \langle A\rangle\langle \overline{A}\rangle \mathsf{be})), \tag{18}$$

$$[U](\mathsf{u} \rightarrow (\neg\langle \overline{A}\rangle * \leftrightarrow \langle \overline{A}\rangle\langle A\rangle \mathsf{eb})), \tag{19}$$

$$[U](\mathsf{u} \rightarrow ((\neg\langle A\rangle * \wedge \neg\langle A\rangle(\mathsf{u} \wedge \langle A\rangle *)) \leftrightarrow \langle A\rangle\langle \overline{A}\rangle \mathsf{eb})). \tag{20}$$

Now, we put

$$\mathsf{CorrBound} = (15) \wedge (16) \wedge (17) \wedge (18) \wedge (19) \wedge (20). \tag{21}$$

Lemma 9. *Let* $\mathbf{M}, [a, b] \Vdash \mathsf{IdDef} \wedge \mathsf{CorrBound}$*, and let* $b_1^0 < b_1^1 < \ldots b_1^{k_1} = b_2^0 < b_2^1 < \ldots < b_2^{k_2} = b_3^0 < \ldots$ *be a sequence of points whose existence is guaranteed by Lemma 7. Then for every* $i \geq 0, j \geq 1$*:*

1. b_j^i *is the beginning point of a* bb*-interval and a* be*-interval if and only if* $1 \leq i \leq k_j - 1$*;*
2. b_j^i *is the beginning point of a* eb*-interval if and only if* $2 \leq i \leq k_j$*;*
3. b_j^i *is the ending point of a* bb*-interval and a* eb*-interval if and only if* $1 \leq i \leq k_j - 2$*;*
4. b_j^i *is the ending point of a* be*-interval if and only if* $2 \leq i \leq k_j - 1$*.*

Proof. Claim (1). By Lemma 7, we know that $\mathbf{M}, [b_j^i, b_j^{i+1}] \Vdash *$ iff $i = 0$. So, if $1 \leq i \leq k_j - 1$, any interval ending in b_j^i is such that the formula $\neg\langle A\rangle *$ is satisfied on it. Therefore, by (15) and of (17), the formulas $\langle \overline{A}\rangle\langle A\rangle \mathsf{bb}$ and $\langle \overline{A}\rangle\langle A\rangle \mathsf{be}$ must be satisfied as well. This means that the point b_j^i is the beginning point of some bb-interval and of some be-interval.
The other claims can be proved by similar arguments. □

Definition 39. *Given two* tile*-intervals* $[c, d]$ *and* $[e, f]$ *in a model* $\mathbf{M}$*,* $[c, d]$ corresponds to $[e, f]$ *if* $\mathbf{M}, [c, e] \Vdash \mathsf{bb}$ *and* $\mathbf{M}, [c, f] \Vdash \mathsf{be}$ *and* $\mathbf{M}, [d, e] \Vdash \mathsf{eb}$.

The following formulas specify the basic relationships between the three types of correspondence:

$$[U] \bigwedge_{c,c' \in \{\mathsf{bb},\mathsf{eb},\mathsf{be}\}, c \neq c'} \neg(c \wedge c'), \tag{22}$$

$$[U](\mathsf{bb} \rightarrow \neg\langle D\rangle\mathsf{bb} \wedge \neg\langle D\rangle\mathsf{eb} \wedge \neg\langle D\rangle\mathsf{be}), \tag{23}$$

$$[U](\mathsf{eb} \rightarrow \neg\langle D\rangle\mathsf{bb} \wedge \neg\langle D\rangle\mathsf{eb} \wedge \neg\langle D\rangle\mathsf{be}), \tag{24}$$

$$[U](\mathsf{be} \rightarrow \langle D\rangle\mathsf{eb} \wedge \neg\langle D\rangle\mathsf{bb} \wedge \neg\langle D\rangle\mathsf{be}). \tag{25}$$

Let us put

$$\mathsf{CorrProp} = (22) \wedge (23) \wedge (24) \wedge (25). \tag{26}$$

Lemma 10. *Let* $\mathbf{M}, [a, b] \Vdash \mathsf{IdDef} \wedge \mathsf{CorrDef} \wedge \mathsf{CorrBound} \wedge \mathsf{CorrProp}$*. Then, for any* $j \geq 1$ *and* $i \geq 1$*:*

1. *The* i*-th tile of the* j*-th* Id*-interval corresponds to the* i*-th tile of the* $j+1$*-th* Id*-interval.*
2. *There are exactly* $j+1$ *tiles in the* $j+1$*-th* Id*-interval.*
3. *No tile of the* j*-th* Id*-interval corresponds to the last tile of the* $j+1$*-th* Id*-interval.*

Proof. To prove the first claim we proceed by nested induction, first on j, then on i.
Let $j = 1$ (base case). The base case of the i-induction directly follows from Lemmas 7, 8, and 9; the inductive step is trivial.
Let $j > 1$ (inductive step). The proof of the base case of the i-induction uses the same argument of the inductive step, but it is simpler than it. Thus, we concentrate our attention on the latter. Let $h > 1$ and suppose (inductive hypothesis) that for all $i < h$, the i-th tile of the j-th Id-interval corresponds to the i-th tile of the $j+1$-th Id-interval.

(bb) Let us show that $[b_j^h, b_{j+1}^h]$ is a bb-interval.
Consider the point b_j^h. By Lemma 9, it must begin some bb-interval that must end at some point c such that $b_{j+1}^1 \leq c \leq b_{j+1}^{k_{j+1}-1}$. Now suppose, for contradiction, that $c \neq b_{j+1}^h$. We must distinguish two cases.
i) Suppose $c < b_{j+1}^h$. Since, by inductive hypothesis, the interval $[b_j^{h-1}, b_{j+1}^h]$ is a be-interval, the bb-interval $[b_j^h, c]$ turns out to be a strict subinterval of such a be-interval, which contradicts $\mathsf{CorrProp}$.
ii) Suppose $c > b_{j+1}^h$. By Lemma 9, the point b_{j+1}^h must end some eb-interval that must begin at some point $d \geq b_j^1$. If $d > b_j^h$, then the eb-interval $[d, b_{j+1}^h]$ is a strict subinterval of the bb-interval $[b_j^h, c]$, which contradicts $\mathsf{CorrProp}$. If $d < b_j^h$, then (by inductive hypothesis) the eb-interval $[b_j^h, b_{j+1}^{h-1}]$ turns out to be a strict subinterval of the eb-interval $[d, b_{j+1}^h]$, which contradicts $\mathsf{CorrProp}$. The last possibility is $d = b_j^h$. By Lemma 9, the point b_{j+1}^h must end some bb-interval $[e, b_{j+1}^h]$, with $e \geq b_j^1$. If $e > b_j^h$, the bb-interval $[e, b_{j+1}^h]$ is a strict subinterval the bb-interval $[b_j^h, c]$, which contradicts $\mathsf{CorrProp}$. If $e = b_j^h$, both bb and be (by the above argument) hold over the interval $[b_j^h, b_{j+1}^h]$, which contradicts $\mathsf{CorrProp}$. Finally, if $e < b_j^h$, the eb-interval $[b_j^h, b_{j+1}^{h-1}]$ is a strict subinterval of the bb-interval $[e, b_{j+1}^h]$, which contradicts $\mathsf{CorrProp}$.
This allows us to conclude that $c = b_{j+1}^h$.

(be) Let us show that $[b_j^h, b_{j+1}^{h+1}]$ is a be-interval.

As in the previous case, by Lemma 9, the point b_j^h must be the beginning point of some be-interval that must end at some point c, with $b_{j+1}^1 \leq c \leq b_{j+1}^{k_{j+1}-1}$. If $c = b_{j+1}^{h+1}$ we are done. Suppose, for contradiction, that $c \neq b_{j+1}^{h+1}$. We must distinguish two cases.

i) Suppose $c < b_{j+1}^{h+1}$. If $c = b_{j+1}^h$, then both bb and be hold over $[b_j^h, b_{j+1}^h]$, which contradicts CorrProp; if $c < b_{j+1}^h$, the be-interval $[b_j^h, c]$ turns out to be a strict subinterval of the be-interval $[b_j^{h-1}, b_{j+1}^h]$, which contradicts CorrProp as well.

ii) Suppose that $c > b_{j+1}^{h+1}$. By Lemmas 9 and 8, the point b_{j+1}^{h+1} must end some be interval $[d, b_{j+1}^{h+1}]$, with $d \geq b_j^1$ (and $d \neq b_j^h$). If $d > b_j^h$, then the be-interval $[d, b_{j+1}^{h+1}]$ is a strict subinterval of the be-interval $[b_j^h, c]$, which contradicts CorrProp. If $d < b_j^h$, the bb-interval $[b_j^h, b_{j+1}^h]$ is a strict subinterval of the be-interval $[d, b_{j+1}^{h+1}]$, which contradicts CorrProp as well.

(eb) Let us show that $[b_j^{h+1}, b_{j+1}^h]$ is a eb-interval.

Consider the point b_{j+1}^h. By Lemma 9, it must be the ending point of some eb-interval $[c, b_{j+1}^h]$, with $b_j^2 \leq c \leq b_j^{k_j}$. If $c = b_j^{h+1}$ we are done. Suppose, for contradiction, that $c \neq b_j^{h+1}$. We must distinguish two cases.

i) Suppose $c > b_j^{h+1}$. By Lemma 9, the point b_j^{h+1} must begin some bb-interval $[b_j^{h+1}, d]$ with $b_{j+1}^1 \leq d \leq b_{j+1}^{k_j-2}$. If $d \leq b_{j+1}^h$, then the bb-interval $[b_j^{h+1}, d]$ is a strict subinterval of the be-interval $[b_j^h, b_{j+1}^{h+1}]$, which contradicts CorrProp. If $d > b_{j+1}^h$, then the eb-interval $[c, b_{j+1}^h]$ is a strict subinterval of the bb-interval $[b_j^{h+1}, d]$, which contradicts CorrProp as well.

ii) Suppose $c < b_j^{h+1}$. If $c = b_j^h$, then both eb and bb (by the previous point) hold over the interval $[b_j^h, b_{j+1}^h]$, which contradicts CorrProp. If $c < b_j^h$, then the eb-interval $[b_j^h, b_{j+1}^{h-1}]$ is a strict subinterval of the eb-interval $[c, b_{j+1}^h]$, which contradicts CorrProp.

As for the second claim, we proceed by induction on j. The base case is straightforward, since, by Lemma 7, there is only one tile for $j = 1$. Suppose now that $j = n$ and for all $l < n$, the l-th ld-interval has exactly l tiles. Assume, for contradiction, that there are $m > n$ tiles in the n-th ld-interval (the case $m < n$ is excluded by the first claim). If $m > n$, the n-th tile is not the last one of the n-th ld-interval and thus, by Lemmas 8 and 9, the point b_n^n must be the ending point of some bb-interval beginning at some point c, with $b_{n-1}^1 \leq c \leq b_{n-1}^{k_{n-1}-1}$. Since $c < b_{n-1}^{k_{n-1}}$, the eb-interval $[b_{n-1}^{k_n}, b_n^{n-1}]$ is a strict subinterval of the bb-interval $[c, b_n^n]$, which contradicts CorrProp. Hence $m = n$.

As for the third claim, suppose that some tile of the j-th ld-interval corresponds to the $j+1$-th tile of the $j+1$-th ld-interval. Then, by definition, $b_{j+1}^{k_{j+1}-1}$ is the ending point of some bb-interval. Since, by Lemma 7, the u-interval $[b_{j+1}^{k_{j+1}}, b_{j+2}^1]$ is a $*$-interval, this contradicts CorrBound (more precisely, formula (16)). □

Encoding the tiling problem. We are now ready to show how to encode the octant tiling problem. First of all, we force the horizontal and the vertical matching of colors by means of the following two formulas:

$$[U]((\mathsf{tile} \wedge \langle A \rangle \mathsf{tile}) \rightarrow \bigvee_{\mathsf{right}(t_i)=\mathsf{left}(t_j)} (T_i \wedge \langle A \rangle T_j)), \quad (27)$$

$$[U](\mathsf{bb} \rightarrow \bigvee_{\mathsf{up}(t_i)=\mathsf{down}(t_j)} (\langle\overline{A}\rangle\langle A\rangle T_i \wedge \langle A\rangle T_j)). \tag{28}$$

Given the set of tiles $\mathcal{T} = \{T_1, \ldots, T_k\}$, we define

$$\Phi_{\mathcal{T}} = \mathsf{IdDef} \wedge \mathsf{CorrDef} \wedge \mathsf{CorrBound} \wedge \mathsf{CorrProp} \wedge (27) \wedge (28). \tag{29}$$

Theorem 53. *Given any finite set of tiles $\mathcal{T} = \{T_1, \ldots, T_k\}$, the formula $\Phi_{\mathcal{T}}$ is satisfiable if and only if $\mathcal{T}$ can tile the second octant $\mathcal{O}$.*

Proof. (Only if:): Suppose that $\mathbf{M}, [a,b] \Vdash \Phi_{\mathcal{T}}$. Then, there is a sequence of points $a = b_1^0 < b_1^1 < b_1^2 = b_2^0 < \ldots b_j^0 < b_j^1 < \ldots < b_j^{j+1} < \ldots$ that satisfies the claims of Lemmas 7, 8, 9, 10. In particular, for every i, j, with $i \leq j$, we have $\mathbf{M}, [b_j^i, b_j^{i+1}] \Vdash \mathsf{tile}$ and hence $\mathbf{M}, [b_j^i, b_j^{i+1}] \Vdash T_k$ for a unique k. We put $f(i,j) = T_k$. From Lemma 10 (and formulas 27 and 28), it follows that the function $f : \mathcal{O} \mapsto \mathcal{T}$ defines a correct tiling of $\mathcal{O}$.

(If:) Let $f : \mathcal{O} \mapsto \mathcal{T}$ be a tiling function. We show that there exist a model $\mathbf{M}$ and an interval $[a,b]$ such that $\mathbf{M}, [a,b] \Vdash \Phi_{\mathcal{T}}$. Let $\mathbf{M} = \langle \mathbb{I}(\mathbb{N}), V \rangle$ be a model whose valuation function V is defined as follows. First of all, for each $i, j \in \mathbb{N}$, we put:

$$\mathsf{u} \in V([i,j]) \Leftrightarrow 0 \leq i = j - 1,$$

which guarantees that (2) is satisfied. Now, let $g : \mathbb{N} \mapsto \mathbb{N}$ be such that $g(n) = (n+1)(n+2)/2 - 1$. For each $(i,j) \in \mathbb{N}$, if $i \leq j$, then

$$* \in V([g(j), g(j)+1]),$$

and

$$f(i,j), \mathsf{tile} \in V([g(j)+i+1, g(j)+i+2]).$$

Tiles and $*$s are assigned to unit intervals only and thus (4) is satisfied too. Since $[g(j)+i+1, g(j)+i+2] = [g(j')+i'+1, g(j')+i'+2]$ only if $(i,j) = (i',j')$, no interval is assigned to two different tiles, and thus (5) is satisfied as well.
Now, for each $j \geq 0$, we define

$$\mathsf{Id} \in V([g(j), g(j+1)]).$$

By definition, the symbol $*$ is associated with the first unit interval of every Id-interval, no Id interval properly begins or ends another Id-interval, and every Id-interval is immediately followed by another Id-interval. Hence, the formula IdDef is satisfied over the interval $[0,2]$.
Finally, for every $i \leq j$, we put

$$\mathsf{bb} \in V([g(j)+i+1, g(j+1)+i+1]),$$
$$\mathsf{be} \in V([g(j)+i+1, g(j+1)+i+2]),$$
$$\mathsf{eb} \in V([g(j)+i+2, g(j+1)+i+1]).$$

It is straightforward to check that formulas $\mathsf{CorrBound}$ and $\mathsf{CorrProp}$ are satisfied. Moreover, since f is a tiling function, formulas (27) and (28) are satisfied as well, whence the thesis. □

As a matter of fact, the model construction in the above proof can be carried out on any linear ordering containing an infinite ascending chain of points. Thus, we obtain the following.

Corollary 12. *The satisfiability problem for any extension of* $\mathrm{PNL}^{\pi+}$ *which is expressive enough to define the operator* $\langle D\rangle$*, interpreted in any class of linear orderings containing a linear ordering with an infinite ascending chain, is undecidable.*

In the rest of the section, we briefly illustrate how the various formulas can be adapted to the other cases.

7.2 Other undecidable extensions of $\mathbf{PNL}^{\pi+}$

Undecidability of $\mathbf{PNL}^{\pi+}$ $\langle\overline{D}\rangle$. Consider now any extension of $\mathrm{PNL}^{\pi+}$ featuring the modality $\langle\overline{D}\rangle$ capturing the relation of strict superinterval, which is the inverse of $\langle D\rangle$. To describe u-intervals, we can rewrite (2) as follows:

$$\mathsf{u} \wedge [\overline{A}][\overline{A}][A]\neg\mathsf{u} \wedge [U]((\mathsf{u} \rightarrow (\neg\pi \wedge \langle A\rangle\mathsf{u})) \wedge ((\mathsf{u} \vee \langle A\rangle\mathsf{u}) \rightarrow \neg\langle\overline{D}\rangle\mathsf{u})). \tag{30}$$

Similarly, to describe Id-intervals, we can rewrite (8) as follows:

$$[U](\langle A\rangle * \rightarrow \neg\langle\overline{D}\rangle\mathsf{Id}). \tag{31}$$

The relation between corr-intervals and Id-intervals can be expressed by rewriting (13) as follows:

$$[U](\mathsf{corr} \rightarrow (\neg\mathsf{u} \wedge \neg\pi \wedge \neg\langle\overline{D}\rangle\mathsf{Id} \wedge \neg\langle A\rangle\mathsf{Id} \wedge \neg\langle\overline{A}\rangle\langle A\rangle\mathsf{First})) \tag{32}$$

$$[U](\mathsf{Id} \rightarrow \neg\langle\overline{D}\rangle\mathsf{corr}). \tag{33}$$

Finally, the relations between be, eb, and bb can be expressed by replacing formulas (23), (24), and (25) with the following ones, where the operator $\langle D\rangle$ has been replaced with $\langle\overline{D}\rangle$:

$$[U](\mathsf{be} \rightarrow \neg\langle\overline{D}\rangle\mathsf{bb}), \tag{34}$$

$$[U](\mathsf{eb} \rightarrow \neg\langle\overline{D}\rangle\mathsf{eb} \wedge \langle\overline{D}\rangle\mathsf{be}), \tag{35}$$

$$[U]((\mathsf{be} \vee \mathsf{bb}) \rightarrow \neg\langle\overline{D}\rangle\mathsf{be}). \tag{36}$$

Undecidability of $\mathbf{PNL}^{\pi+}$ $\langle D\rangle_\sqsubset$ and $\mathbf{PNL}^{\pi+}$ $\langle\overline{D}\rangle_\sqsubset$. If we replace the modality for the strict subinterval relation $\langle D\rangle$ (resp., superinterval relation $\langle\overline{D}\rangle$) with that for the proper subinterval relation $\langle D\rangle_\sqsubset$ (resp., superinterval relation $\langle\overline{D}\rangle_\sqsubset$) the encoding becomes much simpler. In particular, by using any of these operators, it is easy to express the relations between u-intervals, Id-intervals, and corr-intervals. For example, (13) can be expressed as follows:

$$[U]((\mathsf{corr} \rightarrow \neg\langle D\rangle_\sqsubset\mathsf{Id}) \wedge (\mathsf{Id} \rightarrow \neg\langle D\rangle_\sqsubset\mathsf{corr})), \tag{37}$$

or

$$[U]((\mathsf{corr} \rightarrow \neg\langle\overline{D}\rangle_\sqsubset\mathsf{Id}) \wedge (\mathsf{Id} \rightarrow \neg\langle\overline{D}\rangle_\sqsubset\mathsf{corr})). \tag{38}$$

The remaining formulas can be modified in a similar way. For the encoding of the tiling problem, we do not need three types of correspondence intervals anymore. It suffices to use only one propositional variable bb and to express the relation between different bb-intervals as follows:

$$[U]((\mathsf{bb} \rightarrow \neg\langle D\rangle_\sqsubset\mathsf{bb}), \tag{39}$$

or

$$[U]((\mathsf{bb} \rightarrow \neg\langle\overline{D}\rangle_{\sqsubset}\mathsf{bb}). \tag{40}$$

Undecidability of PNL$^{\pi+}$ $\langle B\rangle\langle\overline{E}\rangle$ and PNL$^{\pi+}$ $\langle E\rangle\langle\overline{B}\rangle$. The remaining two cases, namely, the extensions of PNL$^{\pi+}$ containing at least one of the pairs $\langle B\rangle, \langle\overline{E}\rangle$ and $\langle\overline{B}\rangle, \langle E\rangle$ are symmetric. Let us consider the former. The formula (2) becomes:

$$\mathsf{u} \wedge [\overline{A}][\overline{A}][A]\neg\mathsf{u} \wedge [U](\mathsf{u} \rightarrow \neg\langle B\rangle(\neg\pi \wedge \langle A\rangle\mathsf{u})). \tag{41}$$

Similarly, formula (8) can be rewritten as follows:

$$[U](\mathsf{ld} \rightarrow \neg\langle B\rangle(\neg\pi \wedge \langle A\rangle\mathsf{ld})). \tag{42}$$

To encode the tiling problem, two types of correspondence suffice. Indeed, using bb-intervals we can force the existence of eb-intervals by means of the following formula:

$$[U](\mathsf{bb} \wedge \neg\langle\overline{A}\rangle* \rightarrow \langle B\rangle(\mathsf{eb} \wedge \langle\overline{E}\rangle\mathsf{bb} \wedge [\overline{E}](\langle\overline{A}\rangle\mathsf{u} \rightarrow \neg\langle\overline{E}\rangle\mathsf{bb})) \wedge [B](\langle A\rangle\mathsf{u} \rightarrow \neg\langle B\rangle\mathsf{eb})). \tag{43}$$

The tiling problem can be encoded by specifying suitable conditions on bb-intervals and be-intervals.
By using $\langle B\rangle$ only, one can express the relation between corr-intervals and ld-intervals rewriting (13) as follows:

$$[U](corr \rightarrow (\langle B\rangle(\neg\pi \wedge \langle A\rangle*) \wedge [B](\langle A\rangle* \rightarrow \neg\langle B\rangle\langle A\rangle*) \wedge \neg\langle A\rangle*)). \tag{44}$$

Putting together the above results, we have the following theorem.

Theorem 54. *The satisfiability problem for any extension of* PNL$^{\pi+}$ *expressive enough to define one of the following combinations of modal operators:* $\langle D\rangle$, $\langle\overline{D}\rangle$, $\langle D\rangle_{\sqsubset}$, $\langle\overline{D}\rangle_{\sqsubset}$, $\langle B\rangle$ *and* $\langle\overline{E}\rangle$, *and* $\langle\overline{B}\rangle$ *and* $\langle E\rangle$, *interpreted in any class of linear orderings containing a linear ordering with an infinite ascending chain, is undecidable.*

Notice that in most of the considered extensions the inclusion of π is not necessary, since it is definable in the language (this is the case, for instance, when $\langle B\rangle$ belongs to the language). It immediately follows that in such cases the corresponding extensions of the language of PNL^{+} are undecidable as well. The remaining cases, as well as PNL^{-} extensions, are still open.

8 Concluding remarks

We have explored expressiveness and decidability issues for a variety of propositional interval neighborhood logics. First, we have compared PNL$^{\pi+}$ with PNL^{+} and PNL^{-}, and we have shown that the former is strictly more expressive than the other two, which are in a sense incomparable. Then, we have proved that PNL$^{\pi+}$ is decidable by embedding it into FO$^2[<]$ and it is essentially the maximal fragment of HSwith that property. Furthermore, we have proved that PNL$^{\pi+}$ is as expressive as FO$^2[<]$. Finally, we have proved that most extensions of PNL$^{\pi+}$ with other interval modalities are undecidable.
A number of questions still remain open. The most important ones are:

1. Is the satisfiability problem for $\mathrm{PNL}^{\pi+}$ in the classes of all discrete, all dense, or all Dedekind complete linear orders decidable?
2. Is there any decidable extension of PNL with modality from the set $\{\langle B\rangle, \langle E\rangle, \langle O\rangle\}$?

Various natural further developments can stem from the present work. In particular, the tableau systems that have been developed in [18,20,21] for PNL on specific structures such as $\mathbb{N}$ and $\mathbb{Z}$, can be considered for adaptation to deal with $\mathrm{FO}^2[<]$ on these and related classes of linear orders.

References

1. A.A. Aaby and K.T. Narayana. Propositional temporal interval logic is pspace complete. In *Proc. of 9th Inter. Conf. Automated Deduction*, volume 193 of *LNCS*, pages 218–237. Springer, 1985.
2. J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
3. J.F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.
4. J.F. Allen and P. Hayes. A common-sense theory of time. In *Proc. of the 9th International Joint Conference on Artificial Intelligence*, pages 528–531. Morgan Kaufmann, 1985.
5. R. Barua, S. Roy, and C. Zhou. Completeness of neighbourhood logic. *Journal of Logic and Computation*, 10(2):271–295, 2000.
6. R. Barua and C. Zhou. Neighbourhood logics: NL and NL 2. Technical Report 120, UNU/IIST, Macau, 1997.
7. J. Van Benthem. *The Logic of Time (2nd Edition)*. Kluwer Academic Press, 1991.
8. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives of Mathematical Logic. Springer, 1997.
9. H. Bowman, H. Cameron, P. King, and S. Thompson. Specification and prototyping of structured multimedia documents using interval temporal logic. In H. Barringer, M. Fisher, D. Gabbay, and G.Gough, editors, *Advances in Temporal Logic*, volume 16 of *Applied Logic*, pages 435–453. Kluwer Academic, 2000.
10. H. Bowman and S. Thompson. A tableau method for Interval Temporal Logic with projection. In *Proc. of the International Conference Tableaux 1998*, number 1397 in LNCS, pages 108–134. Springer, 1998.
11. H. Bowman and S. Thompson. A decision procedure and complete axiomatization of finite interval temporal logic with projection. *Journal of Logic and Computation*, 13(2):195–239, 2003.
12. D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based Decision Procedure for the Logic of Proper Subinterval Structures over Dense Orderings. In C. Areces and S. Demri, editors, *Proceedings of M4M-5: 5th International Workshop on Methods for Modalities*, pages 335–351, 2007.
13. D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau Systems for Logics of Subinterval Structures over Dense Orderings. In N. Olivetti, editor, *Proc. of TABLEAUX 2007*, volume 4548 of *LNAI*, pages 73–89. Springer, 2007.
14. D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation*, 2008. Accepted for publication.
15. D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. On Decidability and Expressiveness of Propositional Interval Neighborhood Logics. In *Proc. of the International Symposium on Logical Foundations of Computer Science (LFCS)*, volume 4514 of *LNCS*, pages 84–99. Springer, 2007.

16. D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional Interval Neighborhood Logics: Expressiveness, Decidability, and Undecidable Extensions. Technical Report 05, Department of Mathematics and Computer Science, University of Udine, Italy, 2008.
17. D. Bresolin and A. Montanari. A tableau-based decision procedure for a branching-time interval temporal logic. In H. Schlingloff, editor, *Proc. of the 4th Int. Workshop on Methods for Modalities*, pages 38–53, Berlin, Germany, December 2005.
18. D. Bresolin and A. Montanari. A tableau-based decision procedure for Right Propositional Neighborhood Logic. In *Proceedings of TABLEAUX 2005*, volume 3702 of *LNAI*, pages 63–77. Springer, 2005.
19. D. Bresolin and A. Montanari. A tableau-based decision procedure for Right Propositional Neighborhood Logic. Technical Report 02, Department of Mathematics and Computer Science, University of Udine, Italy, 2005.
20. D. Bresolin, A. Montanari, and P. Sala. An optimal tableau-based decision algorithm for Propositional Neighborhood Logic. In *Proceedings of STACS 2007*, volume 4393 of *LNCS*, pages 549–560. Springer, 2007.
21. 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.
22. J. Burgess. Axioms for tense logic, ii. Time periods. *Notre Dame Journal of Formal Logic*, 23(4):375–383, 1982.
23. S. Campos and O. Grumberg. Selective quantitative analysis and interval model checking: Verifying different facets of a system. In R. Alur and T.A. Henzinger, editors, *Proc. of the 8th International Conference on Computer Aided Verification*, number 1102 in LNCS, pages 257–268. Springer, 1996.
24. S. Cerrito, M. Cialdea Mayer, and S. Praud. First-order linear temporal logic over finite time structures. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning*, volume 1705 of *LNAI*, pages 62–76. Springer, 1999.
25. A. Chandra, J. Halpern, A. Meyer, and R. Parikh. Equations between regular terms and an application to process logic. *SIAM Journal on Computing*, 14(4):935–942, 1985.
26. N. Chetcuti-Serandio and L. Fariñas del Cerro. A mixed decision method for duration calculus. *Journal of Logic and Computation*, 10:877–895, 2000.
27. L. Chittaro and A. Montanari. Temporal representation and reasoning in artificial intelligence: Issues and approaches. *Annals of Mathematics and Artificial Intelligence*, 28(1–4):47–106, 2000.
28. M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors. *Handbook of Tableau Methods.* Kluwer, Dordrecht, 1999.
29. L.K. Dillon, G. Kutty, L. E. Moser, P.M. Melliar-Smith, and Y. S. Ramakrishna. Graphical specifications for concurrent software systems. In *Proc. of the 14th International Conference on Software Engineering*, pages 214–224. ACM Press, 1992.
30. L.K. Dillon, G. Kutty, L.E. Moser, P. M. Melliar-Smith, and Y.S. Ramakrishna. An automata–theoretic decision procedure for future interval logic. In Rudrapatna Shyamasundar, editor, *Proc. of the 12th Foundations of Software Technology and Theoretical Computer Science*, volume 652 of *LNCS*, pages 51–67. Springer, 1992.
31. L.K. Dillon, G. Kutty, L.E. Moser, P. M. Melliar-Smith, and Y.S. Ramakrishna. A real-time interval logic and its decision procedure. In *Proc. of the 13th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 761 of *LNCS*, pages 173–192. Springer, 1993.
32. L.K. Dillon, G. Kutty, L.E. Moser, P. M. Melliar-Smith, and Y.S. Ramakrishna. A graphical interval logic for specifying concurrent systems. *ACM Transactions on Software Engineering and Methodology*, 3(2):131–165, 1994.
33. L.K. Dillon, P.M. Melliar-Smith, L.E. Moser, G. Kutty, and Y.S. Ramakrishna. Interval logics and their decision procedures. Part I: an interval logic. *Theoretical Computer Science*, 166(1–2):1–47, 1996.
34. L.K. Dillon, P.M. Melliar-Smith, L.E. Moser, G. Kutty, and Y.S. Ramakrishna. Interval logics and their decision procedures. Part II: a real-time interval logic. *Theoretical Computer Science*, 170(1–2):1–46, 1996.
35. D. Dowty. *Word Meaning and Montague Grammar.* Reidel, Dordrecht, 1979.
36. Z.H. Duan. *An Extended Interval Temporal Logic and A Framing Technique for Temporal Logic Programming.* PhD thesis, University of Newcastle Upon Tyne, 1996.
37. B. Dutertre. Complete proof systems for first order interval temporal logic. *Proc. of the 10th International Symposium on Logic in Computer Science*, pages 36–43, 1995.
38. B. Dutertre. On first order interval temporal logic. Technical report, Royal Holloway, University of London, Technical Report CSD-TR-94-3, 1995.
39. E.A. Emerson. Temporal and modal logic. *Handbook of Theoretical Computer Science*, B:996–1072, 1990.

40. E.A. Emerson and J. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.
41. E.A. Emerson and A.P. Sistla. Deciding full branching time logic. *Information and Control*, 61(3):175–201, 1984.
42. K. Etessami, M. Y. Vardi, and T. Wilke. First-order logic with two variables and unary temporal logic. *Information and Computation*, 179(2):279–295, 2002.
43. M. Franceschet. *Dividing and Conquering the Layered Land*. PhD thesis, Department of Mathematics and Computer Science, University of Udine, Italy, 2002. PhD Thesis Series CS 2002/2.
44. D. M. Gabbay, I. M. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.
45. D.M. Gabbay, M. Reynolds, and M. Finger. *Temporal Logic, vol. 2: Mathematical Foundations and Computational Aspects*. Oxford Science Publications, 2000.
46. A. Galton. A critical examination of Allen's theory of action and time. *Artificial Intelligence*, 42:159–198, 1990.
47. G. Gargov and V. Goranko. Modal logics with names. *Journal of Philosophical Logic*, 22(6):607–636, 1993.
48. V. Goranko, A. Montanari, and G. Sciavicco. A general tableau method for propositional interval temporal logic. In *Proc. of TABLEAUX 2003*, volume 2796 of *LNAI*, pages 102–116, Rome, Italy, September 2003. Springer.
49. V. Goranko, A. Montanari, and G. Sciavicco. Proof systems for propositional interval neighborhood logics. In C. Araces and P. Blackburn, editors, *Proceedings of M4M 3: 3rd International Workshop on Methods for Modalities*, pages 125–139, 2003.
50. V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.
51. V. Goranko, A. Montanari, and G. Sciavicco. A road map on interval temporal logics and duration calculi. In V. Goranko and A. Montanari, editors, *Proc. of the ESSLLI Workshop on Interval Temporal Logics and Duration Calculi*, pages 1–40, 2003.
52. V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *J. of Applied Non-Classical Logics*, 14(1–2):9–54, 2004.
53. V. Goranko, A. Montanari, G. Sciavicco, and P. Sala. A general tableau method for propositional interval temporal logics: Theory and implementation. *Journal of Applied Logic*, 4(3):305–330, 2006.
54. V. Goranko and M. Otto. Model theory of modal logic. In P. Blackburn et al., editor, *Handbook of Modal Logic*, pages 249–329. Elsevier, 2007.
55. 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.
56. D.P. Guelev. A complete fragment of higher-order duration μ-calculus. *Foundations of Software Technology and Theoretical Computer Science*, 20, 2000.
57. D.P. Guelev. Probabilistic neighbourhood logic. In *Proc. of the 6th International Symposium, Pune, India*, volume 1926 of *LNCS*, pages 264–275. Springer, 2000.
58. J.Y. Halpern, Z. Manna, and B. Moszkowski. A hardware semantics based on temporal intervals. In *Proc. of the 10th International Colloquium on Automata, Languages and Programming*, number 154 in LNCS, pages 278–291. Springer, 1983.
59. J.Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
60. C. Hamblin. Instants and intervals. In J.T. Fraser, F.C. Haber, and G.H. Mueller, editors, *The Study of Time (Volume 1)*, pages 324–331. Springer, 1972.
61. M. Hansen. Duration calculus (extended abstract). In V. Goranko and A. Montanari, editors, *Proc. of the ESSLLI Workshop on Interval Temporal Logics and Duration Calculi*, pages 66–84, 2003.
62. M. Hansen and C. Zhou. Duration calculus: Logical foundations. *Formal Aspects of Computing*, 9:283–330, 1997.
63. M. R. Hansen and C. Zhou. Semantics and completeness of Duration Calculus. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, number 600 in LNCS, pages 209–225. Springer, 1992.
64. M.R. Hansen. Model-checking discrete duration calculus. *Formal Aspects of Computing*, 6(6A):826–845, 1994.
65. I.M. Hodkinson, A. Montanari, and G. Sciavicco. Axiomatizability and undecidability of interval temporal logics with C, D, and T. In *Proceedings of CSL 2008: 17th EACSL Annual Conference on Computer Science Logic*, Bertinoro, Italy, 2008.
66. J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

67. L. Humberstone. Interval semantics for tense logics: some remarks. *Journal of Philosophical Logic*, 8:171–196, 1979.
68. D. Van Hung and D.P. Guelev. Completeness and decidability of a fragment of duration calculus with iteration. In P.S. Thiagarajan and R. Yap, editors, *Advances in Computing Science*, volume 1742 of *LNCS*, pages 139–150. Springer, 1999.
69. N. Immerman and D. Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, 1989.
70. H. Kamp. Events, instants and temporal reference. In R. Bäuerle, U. Egli, and A. von Stechov, editors, *Semantics from Different Points of View*, pages 376–417. Springer, 1979.
71. Y. Kesten, Z. Manna, H. McGuire, and A. Pnueli. A decision algorithm for full propositional temporal logic. In *Proc. of the 5th International Conference on Computer Aided Verification*, volume 697 of *LNCS*, pages 97–109, 1993.
72. S. Kono. A combination of clausal and non-clausal temporal logic programs. In M. Fisher and R. Owens, editors, *Executable Modal and Temporal Logics*, number 897 in LNCS, pages 40–57. Springer, 1995.
73. P. Ladkin. *The Logic of Time Representation.* PhD thesis, University of California, Berkeley, 1987.
74. K. Lodaya. Sharpening the undecidability of interval temporal logic. In *Proc. of 6th Asian Computing Science Conference*, number 1961 in LNCS, pages 290–298. Springer, 2000.
75. R.D. Maddux. Relation algebras of every dimension. *Journal of Symbolic Logic*, 57(4):1213–1229, 1992.
76. M. Marx and M. Reynolds. Undecidability of compass logic. *Journal of Logic and Computation*, 9(6):897–914, 1999.
77. M. Marx and Y. Venema. *Multi-Dimensional Modal Logics.* Kluwer Academic Press, 1997.
78. A. Montanari. *Metric and Layered Temporal Logic for Time Granularity.* PhD thesis, Institute for Logic, Language, and Computation, University of Amsterdam, 1996.
79. A. Montanari. Propositional interval temporal logics: some promising paths. In *Proc. of the 12th International Symposium on Temporal Representation and Reasoning (TIME)*. IEEE Computer Society Press, 2005.
80. A. Montanari and G. Sciavicco. A decidable logic for time intervals: Propositional Neighborhood Logic. In F. Anger, G. Ligozat, and H. Guesgen, editors, *Proc. of the AAAI-2002 Workshop on Spatial and Temporal Reasoning*, 2002.
81. A. Montanari and G. Sciavicco. A decidability proof for propositional neighborhood logic (extended abstract). Contributed Talk, Trends in Logics III Conference, Warsaw - Ruciane Nida (Poland), 2005.
82. A. Montanari, G. Sciavicco, and N. Vitacolonna. Decidability of interval temporal logics over split-frames via granularity. In *Proc. of the 8th European Conference on Logics in AI*, volume 2424 of *LNAI*, pages 259–270. Springer, 2002.
83. M. Mortimer. On languages with two variables. *Zeitschr. f. math. Logik u. Grundlagen d. Math*, 21:135–140, 1975.
84. B. Moszkowski. *Reasoning about Digital Circuits.* PhD thesis, Department of Computer Science, Stanford University, Technical Report STAN-CS-83-970, Stanford, CA, 1983.
85. B. Moszkowski. *Executing Temporal Logic Programs.* Cambridge University Press, Cambridge, 1986.
86. B. Moszkowski. Some very compositional temporal properties. In E.R. Olderog, editor, *Programming Concepts, Methods and Calculi*, pages 238–245. IFIP Transaction, A-56, Elsevier Science B.V. (North-Holland), 1994.
87. B. Moszkowski. Compositional reasoning using interval temporal logic and tempura. In W.P. de Roever, H. Langmaak, and A. Pnueli, editors, *Compositionality: the Significant Difference*, number 1536 in LNCS, pages 439–464. Springer, 1998.
88. B. Moszkowski. An automata-theoretic completeness proof for interval temporal logic. In *Proc. of the 27th International Colloquium on Automata, Languages and Programming*, number 1853 in LNCS, pages 223–234. Springer, 2000.
89. B. Moszkowski. A hierarchical completeness proof for interval temporal logic with finite time (preliminary version). In V. Goranko and A. Montanari, editors, *Proc. of the ESSLLI Workshop on Interval Temporal Logics and Duration Calculi*, pages 41–65, 2003.
90. B. Moszkowski and Z. Manna. Reasoning in interval temporal logic. In E. M. Clarke and D. Kozen, editors, *Proc. of the 1983 Workshop on Logics of Programs*, number 164 in LNCS, pages 371–381. Springer, 1984.
91. B. Nebel and H.J. Bürckert. Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66, 1995.
92. M. Otto. Two variable first-order logic over ordered domains. *Journal of Symbolic Logic*, 66(2):685–702, 2001.

93. P.K. Pandya. Weak chop inverses and liveness in mean-value calculus. In *Proc. of the 4th International Conference on Formal Techniques in Real-Time and Fault-Tolerant Systems*, number 1135 in LNCS, pages 148–167. Springer, 1996.
94. J. Penix, C. Pecheur, and K. Havelund. Using model checking to validate AI planner domain models. In *23rd Annual Software Engineering Workshop, NASA Goddard*, 1998.
95. A. Rabinovich. Non-elementary lower bound for propositional duration calculus. *Information Processing Letters*, 66:7–11, 1998.
96. Y.S. Ramakrishna. *Interval Logics for Temporal Specification and Verification.* PhD thesis, University of California, Santa Barbara, 1993.
97. T. Rasmussen. A sequent calculus for signed interval logics. Technical report, 2001.
98. T.M. Rasmussen. Signed interval logic. In *Proc. of the Annual European Conference on Computer Science Logic*, number 1683 in LNCS, pages 157–171. Springer, 1999.
99. T.M. Rasmussen. Labelled natural deduction for interval logics. In *Proc. of the Annual European Conference on Computer Science Logic*, number 2142 in LNCS, pages 308–323. Springer, 2001.
100. H. Reichenbach. *Elements of Symbolic Logic.* The Free Press, New York, 1947.
101. B. Richards and I.-M. Bethke. The temporal logic iq. In J. Oberlander, editor, *Temporal Reference and Quantification: an IQ Perspective.* Centre for Congnitive Science, Univ. of Edinburgh, 1987.
102. P. Roeper. Intervals and tenses. *Journal of Philosophcal Logic*, 9:451–469, 1980.
103. S. Roy. Notes on neighbourhood logic. Technical Report 97, UNU/IIST Technical Report, 1997.
104. P.H. Schmitt and J. Goubault-Larrecq. A tableau system for linear-time temporal logic. In E. Brinksma, editor, *3rd Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 130–144. Lecture Notes in Computer Science 1217, Springer, 1997.
105. D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:377, 1962.
106. I. Shapirovsky and V. Shehtman. Chronological future modality in Minkowski spacetime. In P. Balbiani, N. Y. Suzuki, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 4, pages 437–459. King's College Publications, London, 2003.
107. E.V. Sørensen, A.P. Ravn, and H. Rischel. Control program for a gas burner. Part i: Informal requirements, process case study 1. Technical report, ProCoS Report ID/DTH EVS2, 1990.
108. P. van Beek. Approximation algorithms for temporal reasoning. In *Proc. of the 11th International Joint Conference on Artificial Intelligence*, pages 1291–1296. Morgan Kaufmann, 1989.
109. Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.
110. Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.
111. M.B. Vilain and H. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proc. of the 5th National Conference of the American Association for Artificial Intelligence (AAAI)*, pages 377–382. AAAI Press, 1986.
112. P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 28:119–136, 1985.
113. C. Zhou. Duration calculus, a logical approach to real-time systems. In *Proc. of the Annual Conference of the European Association for Computer Science Logic*, number 1548 in LNCS, pages 1–7. Springer, 1999.
114. C. Zhou and M. R. Hansen. An adequate first order interval logic. In W.P. de Roever, H. Langmaak, and A. Pnueli, editors, *Compositionality: the Significant Difference*, number 1536 in LNCS, pages 584–608. Springer, 1998.
115. C. Zhou, M.R. Hansen, and P. Sestoft. Decidability and undecidability results for duration calculus. In *Proc. of the 10th Symposium on Theoretical Aspects of Computer Science*, number 665 in LNCS, pages 58–68. Springer, 1993.
116. C. Zhou, C.A.R. Hoare, and A. P. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269–276, 1991.
117. C. Zhou and L. Xiaoshan. A mean value calculus of durations. In A. W. Roscoe, editor, *A Classical Mind: Essays in Honour of C.A.R. Hoare*, pages 431–451. Prentice Hall International Series in Computer Science, 1994.