

UNIVERSITÀ DEGLI STUDI DI UDINE
DIPARTIMENTO DI MATEMATICA E INFORMATICA
DOTTORATO DI RICERCA IN INFORMATICA

PH.D. THESIS

Proof Methods for Interval Temporal Logics

CANDIDATE:
Davide Bresolin

SUPERVISOR:
Prof. Angelo Montanari

Author's e-mail: bresolin@dimi.uniud.it

Author's address:

Dipartimento di Matematica e Informatica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

*With magic you can turn a frog into a prince.
With science you can turn a frog into a PhD and you
still have the frog you started with.*

The Science Of Discworld, Pratchett, Stewart, and Cohen

Contents

Introduction	iii
1 The Framework of Interval Logics	1
1.1 Structures for Time Intervals	2
1.2 Relations between intervals	4
1.3 Some Propositional Interval Logics	5
1.3.1 The logic HS	6
1.3.2 The logics CDT and BCDT ⁺	7
1.3.3 The logic BE	8
1.3.4 The sub-interval Logic D	9
1.3.5 The logics B \bar{B} and E \bar{E}	10
1.3.6 Split Logics	10
1.4 Neighborhood Logics	12
1.4.1 Propositional Neighborhood Logic	12
1.4.2 Right Propositional Neighborhood Logic	13
1.4.3 Branching Time Neighborhood Logic	13
2 Tableaux and Dual-tableaux for Temporal Logics	15
2.1 Tableaux for point-based temporal logics	16
2.1.1 Tableaux for Linear Temporal Logic	16
2.1.2 Tableaux for Computational Tree Logic	18
2.2 Tableaux for interval-based temporal logics and Duration Calculi	18
2.2.1 A tableau for LPITL _{proj}	19
2.2.2 A tableau for Propositional Duration Calculus	20
2.2.3 A tableau for BCDT ⁺	21
2.3 Dual-tableaux for standard relational logic	24
3 The tableau method for RPNL	27
3.1 An intuitive account of the method	27
3.2 Labelled Interval Structures and satisfiability	30
3.3 The complexity of the satisfiability problem for RPNL ^{π^+}	35
3.3.1 An upper bound to the computational complexity	35
3.3.2 A lower bound to the computational complexity	37
3.4 A tableau-based decision procedure for RPNL ^{π^+}	41
3.4.1 The tableau method	41
3.4.2 Computational complexity	43
3.4.3 Soundness and completeness	45
3.5 A tableau-based decision procedure for RPNL ⁺ and RPNL ⁻	48

4	The tableau method for BTNL	49
4.1	Basic Notions	49
4.2	Tableau construction	51
4.3	The decision procedure	56
4.4	The decision procedure at work	57
5	The tableau method for PNL	61
5.1	Labelled Interval Structures for PNL	61
5.2	A tableau-based decision procedure for $\text{PNL}^{\pi+}$	67
5.2.1	Soundness and completeness	70
5.2.2	Computational complexity	74
5.3	A tableau for PNL^+ and PNL^-	74
6	Decidability and expressiveness of PNL	77
6.1	The two-variable fragment of first-order logic	78
6.2	Comparing the expressive power of logics	79
6.3	$\text{PNL}^{\pi+}$, PNL^+ , and PNL^- expressiveness	80
6.4	Decidability of PNL	83
6.5	Expressive Completeness	84
6.6	$\text{PNL}^{\pi+}$ and other HS fragments	87
7	A relational approach to interval logics	89
7.1	A relational logic for HS	90
7.2	Translation	92
7.3	The proof system for RL_{HS}	94
7.3.1	Decomposition rules	94
7.3.2	Specific rules	95
7.3.3	Axiomatic sets	95
7.3.4	Proof trees and soundness of the proof system	96
7.3.5	Completion conditions	97
7.3.6	Branch model	98
7.4	HS-validity and RL_{HS} -provability	102
7.5	Extensions of the relational system	103
7.5.1	Incorporating the other interval relations	103
7.5.2	Considerations on the nature of intervals	106
7.5.3	Properties of the temporal ordering	107
	Conclusions	111
	Bibliography	117

Introduction

The aim of this dissertation is to explore the use of *intervals* as a formal tool for modeling and reasoning about time in logics and computer science.

Temporal reasoning has been devoted great attention in computer science since it has been proved useful in the specification and verification of programs and reactive systems [Pnu77]. In this context, time is viewed as a linear (and often discrete) sequence of points representing the evolution of the system, or as a branching structure where every path starting at a given point d is a possible evolution from the state represented by d . *Linear Temporal Logic* (LTL) and *Computational Tree Logic* (CTL) are well-known modal logics that correspond to such models of time and that have been deeply investigated in the literature [Eme90].

In this dissertation we follow an alternative approach, and we consider time as a set of *intervals*, that is, periods of time with a duration. In many contexts, such a notion of time provides a more intuitive, adequate and compact description of the considered portion of reality.

In [All83], Allen compares time points and intervals as basic ontologies for talking about temporal events, and the result is definitely in favour of the latter approach. Real world events have a *duration*, and thus “durationless” time points cannot deal very well with them. Conversely, intervals can support variations of the grain of reasoning and relative imprecision of temporal information (e.g., an event may end before another, but the exact relationship between them may be unknown) in a way that a model of time based on points cannot support. Another detailed comparison and analysis of point based and interval based time models is made in [vB91]. It turns out that in the case of intervals several philosophical and logical paradoxes disappear: as an example, Zeno’s flying arrow paradox (“if at each *instant* the flying arrow stands still, how is movement possible?”) and the dividing instant dilemma (“if the light is on and it is turned off, what is its state at the *instant* between the two events?”). In such cases, a point based time model cannot represent correctly the way we perceive reality.

The need for interval based temporal representations of knowledge arises in many fields of computer science, such as artificial intelligence, planning, database systems, and computational linguistics. The use of temporal intervals as a formalism for specifying and verifying hardware systems and programs was first studied in [Mos83] and [HMM83], where the authors proposed a logic, called *Interval Temporal Logic* (ITL), that is an extension of LTL where formulae are evaluated over sequences of states (intervals) instead of single states. In such an approach, the behavior of hardware and software systems is decomposed into successively smaller periods of activity, and ITL provides a convenient language to specify and analyze such a behavior. Other interval based logics that have been proposed in the literature are

Halpern and Shoham’s Modal Logic of Time Intervals (HS) [HS91], Venema’s CDT logic [Ven91], and Goranko, Montanari, and Sciavicco’s Propositional Neighborhood Logic (PNL) [GMS03b] (an up-to-date survey of the field can be found in [GMS04]).

Unfortunately, in the area of interval logics, undecidability is the rule and decidability the exception, and most such logics turned out to be (highly) undecidable. Interval logics make it possible to express properties of *pairs* of time points (think of intervals as constructed out of points), rather than *single* time points. In most cases, this feature precludes the possibility of reducing interval-based temporal logics to point-based ones. However, there are a few exceptions where the logic satisfies suitable *syntactic and/or semantic restrictions*, and such a reduction can be defined, thus allowing one to benefit from the good computational properties of point-based logics [Mon05].

One can get decidability by making a suitable choice of the interval modalities. This is the case with the $\langle B \rangle \langle \bar{B} \rangle$ and $\langle E \rangle \langle \bar{E} \rangle$ fragments of HS. Given a formula ϕ and an interval $[d_0, d_1]$, $\langle B \rangle \phi$ (resp. $\langle \bar{B} \rangle \phi$) holds over $[d_0, d_1]$ if ϕ holds over $[d_0, d_2]$, for some $d_0 \leq d_2 < d_1$ (resp. $d_1 < d_2$), and $\langle E \rangle \phi$ (resp. $\langle \bar{E} \rangle \phi$) holds over $[d_0, d_1]$ if ϕ holds over $[d_2, d_1]$, for some $d_0 < d_2 \leq d_1$ (resp. $d_2 < d_1$). Consider the case of $\langle B \rangle \langle \bar{B} \rangle$ (the case of $\langle E \rangle \langle \bar{E} \rangle$ is similar). As shown by Goranko et al. [GMS04], the decidability of $\langle B \rangle \langle \bar{B} \rangle$ can be obtained by embedding it into the propositional temporal logic of linear time (TL[F,P]) with temporal modalities F (sometime in the future) and P (sometime in the past). The formulae of $\langle B \rangle \langle \bar{B} \rangle$ are simply translated into formulae of TL[F,P] by a mapping that replaces $\langle B \rangle$ by P and $\langle \bar{B} \rangle$ by F . TL[F,P] has the finite model property and it 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) investigated by Montanari et al. in [MSV02]. 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 the 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).

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 variable 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 if it is true over all its subintervals (*homogeneity*). A decidable fragment of ITL extended with quantification over propositional variables (QPITL) has been obtained by imposing the *locality* constraint. By exploiting such a constraint, decidability of QPITL can be proved by embedding it into quantified LTL. (In fact, as already noted by Venema, the locality assumption yields decidability even in the case of quite expressive interval logics such as HS and CDT.)

A major challenge in the area of 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.

The main objective of the dissertation will be the exploration of the boundary area between decidability and undecidability in the field of Interval Temporal Logics, by developing decision procedures for interval logics whose decidability is still unknown.

After an introductory part that describes the general framework of interval temporal logics and some of the most interesting interval logics proposed in the literature, this dissertation discusses existing tableau and dual-tableau methods for point-based and interval-based temporal logics.

Chapters 3, 4, 5, and 6 are dedicated to Propositional Neighborhood Logic (PNL). More precisely, in Chapter 3 we present a decision procedure for the future fragment of PNL (RPNL), interpreted over the natural numbers. Then, in Chapter 4 such a decision procedure is extended to an original branching-time temporal logic, called Branching Time Neighborhood Logic (BTNL), that combines CTL operators with the operators of RPNL. In Chapter 5 the decision procedure for RPNL is extended to full PNL interpreted over the integers. We prove the soundness and completeness of these decision procedures, and we study their complexity. Finally, Chapter 6 establishes an interesting connection between PNL and the two-variable fragment of first-order logic extended with a linear ordering. Such a connection allows us to obtain a general decidability result for PNL over various linear orders as well as to get useful insights about the relationships between PNL and other interval-based and point-based logics.

The last part of the thesis (Chapter 7) presents an original relational proof system in the style of dual tableaux for relational logics associated with modal logics of temporal intervals that allows one to prove validity and entailment in several propositional interval logics, interpreted over various classes of linear orderings. Conclusions provide an assessment of the main results of the dissertation and outline future research developments.

1

The Framework of Interval Logics

In many formalization of time, for example in analysis, physics, geometry and in most of the logical approaches, “points” or “instants” are taken as primitive objects. They are usually defined as entities without a duration. However, this concept is not an intuitive one. All temporal entities we experience *have* a duration in time!

In this dissertation we will refer to a more intuitive approach to time, where “objects with a duration”, that we will call “intervals” or “periods”, are taken as primitive, non definable, concepts. The first problem we have to face is how to translate this intuitive notion into a precise mathematical formalism. Two approaches are possible: either intervals are primitive objects of the model and studied on their own, without referring to their internal structure, or they are built up from a traditional point-based temporal domain.

The first approach has been followed by van Benthem in [vB91], where a “reasonable choice of basic principles embodying the minimum conditions for a structure to qualify as a ‘period structure’ ” has been studied and analyzed. The author started from two examples of interval structures, namely, the closed intervals over the integers and the open intervals over the reals, and defined general principles by abstracting from those concrete structures. He considered the relations of inclusion \sqsubseteq (“it is a subinterval of”) and precedence \prec (“it is entirely before”) between intervals, and studied the first-order theory of structures of the form (I, \sqsubseteq, \prec) , where I is simply a non-empty set of atomic objects called “intervals”. This same approach has been followed by Montanari et al. in [MSV02] and by Vitacolonna in [Vit05], where a purely interval theory of a particular class of interval structures (Split Structures) and of interval logics (Split Logics) has been developed (see Section 1.3.6). While this approach seems cleaner from a “philosophical” point of view, it turns out that the basic principles needed to directly define an interval structure are more involved than the ones for point-based structures. Furthermore, many usual properties of flows of time, like linearity, density, and discreteness, that are easily defined in terms of points do not directly transfer to intervals.

The latter approach is the most common in the interval logics literature [GMS04, HS91, Ven91] and the easiest way to define interval structures. In this view, an

underlying *flow of time* is modelled as a strict partial ordering of time points, while intervals are defined as *sets* of time points satisfying some particular constraints. By doing so, all the usual properties of strict orderings (like linearity, density, discreteness, unboundedness, ...) can be easily defined and transferred to interval structures.

In this dissertation we stick to this second approach, and define intervals as pairs of points $[d_0, d_1]$ such that $d_0 \leq d_1$. This chapter is devoted to precisely formalizing the concept of interval structure and to survey the main propositional interval logics proposed in the literature, with respect to their expressivity, axiomatizability, and (un)decidability results.

1.1 Structures for Time Intervals

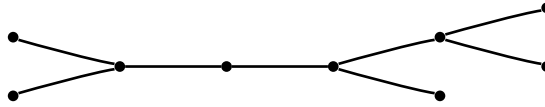
Given a strict partial ordering $\mathbb{D} = \langle D, < \rangle$, an *interval* in \mathbb{D} is a pair $[d_0, d_1]$ such that $d_0, d_1 \in D$ and $d_0 \leq d_1$. $[d_0, d_1]$ is a *strict interval* if $d_0 < d_1$, while intervals of the form $[d_0, d_0]$ are called *point-intervals*. A point $d \in D$ belongs to the interval $[d_0, d_1]$ if $d_0 \leq d \leq d_1$. We denote the set of all strict intervals on \mathbb{D} as $\mathbb{I}(\mathbb{D})^-$, while the set of all (strict and point) intervals on \mathbb{D} will be denoted by $\mathbb{I}(\mathbb{D})^+$. With $\mathbb{I}(\mathbb{D})$ we denote either of these.

Definition 1.1. Given a strict partial ordering $\mathbb{D} = \langle D, < \rangle$ and a set of intervals $\mathbb{I}(\mathbb{D})$, we call a pair $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ an *interval structure*. An interval structure of the form $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle$ (resp. $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$) is called a *strict* (resp. *non-strict*) *interval structure*.

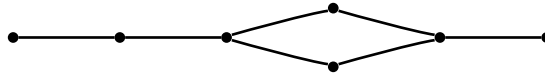
In all interval structures considered in this dissertation, the intervals will be assumed *linear*. Thus, we will concentrate on partial orders with the *linear interval property*:

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

Clearly, every linear order has the linear interval property. An example of a non-linear order with this property is the following:



while a non-example is:



Analogously to point structures, an interval structure can be:

- **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, namely,

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

and

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

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

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

- **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 these interval structures, other interesting ones that will be considered in this dissertation are those based on \mathbb{N} , \mathbb{Z} , \mathbb{Q} , and \mathbb{R} with their usual orderings.

Branching-time structures

A particular class of structures is the one of *branching-time structures*. According to a commonly accepted perspective [Eme90], the underlying temporal structure of branching-time temporal logics has a branching-like nature where each time point may have many successor points. The structure of time thus corresponds to an infinite tree. We shall further assume that the timeline defined by every (infinite) path in the tree is isomorphic to $\langle \mathbb{N}, < \rangle$. We allow a node in the tree to have infinitely many (possibly, uncountably many) successors, while we require each node to have at least one successor. It will turn out that, as far as our logics are concerned, such trees are indistinguishable from trees with finite branching.

Given a directed graph $\mathbb{G} = \langle G, S \rangle$, a *finite S-sequence* over \mathbb{G} is a sequence of nodes $g_1 g_2 \dots g_n$, with $n \geq 2$ and $g_i \in G$ for $i = 1, \dots, n$, such that $S(g_i, g_{i+1})$ for $i = 1, \dots, n - 1$. *Infinite S-sequences* can be defined analogously. We define a *path* ρ in \mathbb{G} as a finite or infinite *S-sequence*. In the following, we shall take advantage of a relation $S^+ \subseteq G \times G$ such that $S^+(d_i, d_j)$ if and only if g_i and g_j are respectively the first and the last element of a finite *S-sequence*.

Temporal structures for branching time logics are infinite trees defined as follows.

Definition 1.2. An infinite *tree* is an infinite directed graph $\mathbb{T} = \langle T, S \rangle$, with a distinguished element $t_0 \in T$, called the *root* of the tree, where T is the set of nodes, called *time points*, and the set of edges $S \subseteq T \times T$ is a relation such that:

- for every $t (\neq t_0) \in T$, $S^+(t_0, t)$, that is, every point is *S-reachable* from the root;

- for every $t(\neq t_0) \in T$, there exists at most one $t' \in T$ such that $S(t', t)$ (together with the previous one, this condition guarantees that every point different from the root has exactly one S -predecessor);
- there exists no t' such that $S(t', t_0)$, that is, t_0 has no S -predecessors;
- for every $t(\neq t_0) \in T$, there exists at least one $t' \in T$ such that $S(t, t')$, that is, every point has at least one S -successor.

It is not difficult to show that infinite trees are *acyclic* graphs, that is, there exist no finite paths which start from and end at the same node.

Given an infinite tree $\mathbb{T} = \langle T, S \rangle$, we can define a partial order $<$ over T such that, for every $t, t' \in T$, $t < t'$ if and only if $S^+(t, t')$. It is immediate that, for every infinite path ρ in \mathbb{T} , $\langle \rho, < \rangle$ is isomorphic to $\langle \mathbb{N}, < \rangle$. Given a tree $\mathbb{T} = \langle T, S \rangle$, and the corresponding partial ordering $\langle T, < \rangle$, we can generalize Definition 1.1 to trees, and we denote the set of all strict intervals on \mathbb{T} as $\mathbb{I}(\mathbb{T})^-$, and the set of all intervals as $\mathbb{I}(\mathbb{T})^+$.

1.2 Relations between intervals

In linear orders, the 13 Allen's binary relations between intervals are usually considered [All83]: *equals*, *ends*, *during*, *begins*, *overlaps*, *meets*, *before* together with their inverses. These relations are graphically represented in Figure 1.1.

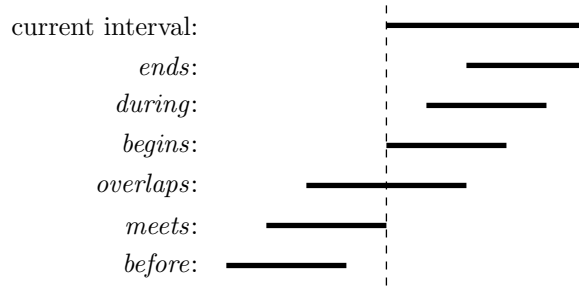


Figure 1.1: Allen's relations

Another natural binary relation between intervals, definable in terms of the Allen's relations, is the *sub-interval* relation. Given a partial ordering \mathbb{D} and two intervals $[s_0, s_1]$ and $[d_0, d_1]$, we have that:

- $[s_0, s_1]$ is a *sub-interval* of $[d_0, d_1]$ (denoted by $[s_0, s_1] \sqsubseteq [d_0, d_1]$), if $d_0 \leq s_0$ and $s_1 \leq d_1$;
- $[s_0, s_1]$ is a *proper sub-interval* of $[d_0, d_1]$ (denoted by $[s_0, s_1] \sqsubset [d_0, d_1]$), 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 by $[s_0, s_1] \sqsubset [d_0, d_1]$), if $d_0 < s_0$ and $s_1 < d_1$.

Notice that the strict sub-interval relation \sqsubset corresponds to Allen's relation *during*, while \sqsubset and \sqsubseteq correspond to *begins* \cup *during* \cup *ends* and *equals* \cup *begins* \cup *during* \cup *ends*, respectively.

As for *ternary* relations between intervals, there is one of particular importance in the area of propositional interval logics. Such a ternary relation is the relation A , that was introduced by Venema in [Ven91], and can be graphically depicted as in Figure 1.2. Formally, we have that $Aijk$ holds if and only if i *meets* j , i *begins* k , and j *ends* k , that is, k is the concatenation of i and j .

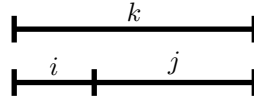


Figure 1.2: Venema's ternary relation A

In Chapters 3, 4, 5, and 6 we will focus our attention on the relation *meets* and its inverse *met*, that we call *left neighborhood* and *right neighborhood* relation, respectively. Interval temporal logics based on these relations are known as *Neighborhood Logics*.

1.3 Some Propositional Interval Logics

In this section we introduce and analyze some well known propositional interval logics.

The generic language of propositional interval logics includes the set of propositional variables AP , the classical propositional connectives \neg and \vee (all others, including the propositional constants \top and \perp , are definable as usual), and a set of *interval temporal operators (modalities)* specific for each logical system.

As pointed out in the previous sections, there are two different natural semantics for interval logics, namely, the *strict* one, which excludes point intervals, and a *non-strict* one, which includes them. A *strict model* for a formula is a tuple $\mathbf{M}^- = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-, \mathcal{V} \rangle$ where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle$ is a strict interval structure with the linear interval property, and $\mathcal{V} : AP \rightarrow 2^{\mathbb{I}(\mathbb{D})^-}$ is the *valuation function* that assigns to every propositional variables p the set of intervals on which p holds. Respectively, a *non-strict model* is a tuple $\mathbf{M}^+ = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$ is a non-strict interval structure with the linear interval property, and $\mathcal{V} : AP \rightarrow 2^{\mathbb{I}(\mathbb{D})^+}$. When we do not wish to specify the strictness, we will simply write \mathbf{M} , assuming either version.

The semantic of interval temporal logics is sometimes subjected to restrictions justified by the specific applications for which a logical system has been designed, as, for example:

- *locality*, meaning that a propositional variable is true over an interval if and only if it is true at its starting point;
- *homogeneity*, requiring that truth of a formula at an interval implies truth of that formula at every sub-interval of it.

A different kind of restriction is imposed by the so-called *split-structures* (see Section 1.3.6), where not all sub-intervals of the current interval are “available”, but only those two which are determined by the “split-point” of the current interval.

Unless otherwise specified, we interpret our interval logics over interval structures where all intervals are present and do not assume any semantic restriction on the valuation of formulae. As an example, given an interval $[d_i, d_j]$, it may happen that $[d_i, d_j] \in \mathcal{V}(p)$ while $[d'_i, d'_j] \notin \mathcal{V}(p)$ for all intervals $[d'_i, d'_j]$ strictly contained in $[d_i, d_j]$.

1.3.1 The logic HS

The most expressive propositional interval logic with unary modal operators studied in the literature is Halpern and Shoham’s logic HS [HS91]. HS contains (as primitive or definable) all unary modalities that correspond to the relations between intervals depicted in Figure 1.1, so it can be considered as the temporal logic of Allen’s relations. It features, as primitives, the modalities $\langle B \rangle$ (*begins*) $\langle E \rangle$ (*ends*), and $\langle A \rangle$ (*met by*), as well as their inverses $\langle \bar{B} \rangle$, $\langle \bar{E} \rangle$, and $\langle \bar{A} \rangle$, which suffices to define all other modal operators. HS was originally interpreted over non-strict models based on partial orders with the linear interval property.

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

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle B \rangle\varphi \mid \langle \bar{B} \rangle\varphi \mid \langle E \rangle\varphi \mid \langle \bar{E} \rangle\varphi \mid \langle A \rangle\varphi \mid \langle \bar{A} \rangle\varphi.$$

The semantics of HS is defined recursively by the *satisfiability relation* \Vdash as follows. Let $\mathbf{M} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{V} \rangle$ be some given model, and let $[d_i, d_j] \in \mathbb{I}(\mathbb{D})$:

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

It is worth noticing that no interval is a beginning/ending interval of itself, and that point intervals have no beginning or ending intervals at all. Hence, on the non-strict semantics, point intervals and strict intervals can be distinguished by the formula $[B] \perp$, which is true only on the former ones.

The *begin point* and *end point* modalities (denoted $\llbracket BP \rrbracket$ and $\llbracket EP \rrbracket$) relate an interval with its beginning and ending point, respectively. A formula $\llbracket BP \rrbracket \varphi$ (resp., $\llbracket EP \rrbracket \varphi$) is true over an interval $[d_i, d_j]$ if and only if φ holds on the interval $[d_i, d_i]$ (resp., $[d_j, d_j]$). They can be defined in HS (over non-strict interval structures) as follows.

$$\begin{aligned} \llbracket BP \rrbracket \varphi &::= ([B] \perp \wedge \varphi) \vee (\langle B \rangle ([B] \perp \wedge \varphi)) \\ \llbracket EP \rrbracket \varphi &::= ([B] \perp \wedge \varphi) \vee (\langle E \rangle ([B] \perp \wedge \varphi)) \end{aligned}$$

In the non-strict semantics, the operators $\langle A \rangle$ and $\langle \bar{A} \rangle$ can be defined using the other ones:

$$\langle A \rangle \varphi ::= \llbracket EP \rrbracket \langle \bar{B} \rangle \varphi \quad \langle \bar{A} \rangle \varphi ::= \llbracket BP \rrbracket \langle \bar{E} \rangle \varphi$$

while in the strict semantics they have to be taken as primitives.

HS is a highly undecidable logic. In [HS91], the authors have obtained important results about non-axiomatizability, undecidability, and complexity of the satisfiability problem for HS in many natural classes of models. It turns out that the validity problem for HS, interpreted over any class of ordered structures with an infinitely ascending sequence is at least r.e.-hard. In the case of Dedekind-complete ordered structures having an infinitely ascending sequence (e.g., natural numbers and reals), it becomes Π_1^1 -hard. Furthermore, it is possible to show that undecidability occurs even without existence of infinitely ascending sequences.

1.3.2 The logics CDT and BCDT⁺

Venema's CDT logic [Ven91] is the most expressive propositional interval logic over linear orderings. An extension of CDT to partial orderings with the linear interval property, called BCDT⁺, has been investigated by Goranko, Montanari and Sciavicco in [GMS03a, GMSS06]. The language of CDT and BCDT⁺ features the modal constant π , that holds only on point intervals,^a and three binary operators C , D , and T , based on the ternary relation A (Figure 1.2). Formulae of CDT are generated by the following abstract syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid \pi \mid \varphi C \varphi \mid \varphi D \varphi \mid \varphi T \varphi.$$

The satisfiability relation \Vdash for the modal operators π , C , D , and T is defined as follows.

- $\mathbf{M}, [d_i, d_j] \Vdash \pi$ iff $d_i = d_j$;
- $\mathbf{M}, [d_i, d_j] \Vdash \varphi C \psi$ iff $\exists d_k \in D$, $d_i \leq d_k \leq d_j$, such that $\mathbf{M}, [d_i, d_k] \Vdash \varphi$ and $\mathbf{M}, [d_k, d_j] \Vdash \psi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \varphi D \psi$ iff $\exists d_k \in D$, $d_k \leq d_i$, such that $\mathbf{M}, [d_k, d_i] \Vdash \varphi$ and $\mathbf{M}, [d_k, d_j] \Vdash \psi$;

- $\mathbf{M}, [d_i, d_j] \Vdash \varphi \ T \ \psi$ iff $\exists d_k \in D, d_k \geq d_j$, such that $\mathbf{M}, [d_j, d_k] \Vdash \varphi$ and $\mathbf{M}, [d_i, d_k] \Vdash \psi$.

As for the relationship with the other propositional interval logics, interpreted over linear ordering, it is immediate that CDT subsumes HS:

$$\begin{aligned} \langle B \rangle \varphi &= \varphi \ C \ (\neg \pi); & \langle \overline{B} \rangle \varphi &= (\neg \pi) \ T \ \varphi; \\ \langle E \rangle \varphi &= (\neg \pi) \ C \ \varphi; & \langle \overline{E} \rangle \varphi &= \varphi \ D \ (\neg \pi); \\ \langle A \rangle \varphi &= (\neg \pi \wedge \varphi) \ T \ \top; & \langle \overline{A} \rangle \varphi &= (\neg \pi \wedge \varphi) \ D \ \top; \end{aligned}$$

Since HS is the propositional interval logic of Allen's relations, every propositional interval logic with unary modalities based on Allen's relations is subsumed by CDT. Furthermore, as a consequence of the previous results for HS, the satisfiability (resp. validity) problem for CDT is not decidable over almost all interesting classes of linear ordering, including \mathbb{N} , \mathbb{Z} , \mathbb{R} , etc.

1.3.3 The logic BE

An interesting fragment of HS is the logic BE, that was first proposed and studied by Lodaya in [Lod00]. BE features only the temporal operators $\langle B \rangle$ and $\langle E \rangle$, and it was originally interpreted over linear, non-strict interval structures. Its syntax is the following.

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

As pointed out in the case of HS, the modal constant π is definable as $[B] \perp$. Hence, the beginning point and ending point modalities $\llbracket BP \rrbracket$ and $\llbracket EP \rrbracket$ can be defined as shown above. Another interesting modality that is definable in BE is the *universal modality* $\llbracket All \rrbracket$, that forces a formula to hold over every interval of the model:

$$\llbracket All \rrbracket \varphi ::= [B] \varphi \wedge [E] \varphi \wedge [B][E] \varphi.$$

Despite its simplicity, BE is expressive enough to capture some relevant classes of interval structures by means of *constraint formulae*. First, one can constrain an interval $[d_0, d_1]$ to be such that $d_0 < d_1$ and there are no points between d_0 and d_1 by means of the following formula:

$$l_1 ::= \langle B \rangle \top \wedge [B][B] \perp.$$

Hence, discreteness can be defined as follows

$$\mathbf{discrete} ::= \llbracket All \rrbracket (\pi \wedge l_1 \wedge (\langle B \rangle l_1 \wedge \langle E \rangle l_1)),$$

while density can be imposed by the following formula

$$\mathbf{dense} ::= \llbracket All \rrbracket \neg l_1.$$

In [Lod00], Lodaya proved that BE is undecidable over non-strict, dense, linear interval structures. Since density is expressible by a constant formula, the satisfiability

of a formula φ in a dense model is equivalent to the satisfiability of $[All]\neg l_1 \wedge \varphi$ in any linear non-strict model. Hence, the satisfiability problem for BE over all non-strict linear orderings is undecidable.

To the best of our knowledge, the decidability of BE over special classes of linear orderings (such as the natural numbers), or over strict models, and the definition of a sound and complete axiomatic system for BE are still open.

1.3.4 The sub-interval Logic D

The logic D is the logic of the sub-interval relation. Since D can only look inside the current interval, from the linear interval property it follows that we can restrict our attention to the class of all linear structures.

The abstract syntax of the logic D is:

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

Besides the strict and non-strict versions, the logic D admits different semantic variations, depending on which sub-interval relation (\sqsubseteq , \sqsubset , or \sqsupseteq) is assumed. Thus, the formal semantics of the operator $\langle D \rangle$ is the following.

- $\mathbf{M}, [d_i, d_j] \Vdash \langle D \rangle \varphi$ iff there exists a sub-interval $[d_k, d_m]$ of $[d_i, d_j]$ such that $\mathbf{M}, [d_k, d_m] \Vdash \varphi$.

In the following, we use D_{\sqsubseteq} , D_{\sqsubset} , and D_{\sqsupseteq} to distinguish among the various semantics of the logic of subintervals.

The sub-interval logic was first studied by van Benthem in [vB91], where the reflexive subinterval relation \sqsubseteq is considered. The author proves that when the strict semantics is considered and formulae are interpreted over the rational numbers, the logic D_{\sqsubseteq} becomes equivalent to the standard modal logic S4 (that is, the logic of reflexive and transitive frames). Since in the case of the reflexive relation \sqsubseteq every subinterval frame is a reflexive and transitive frame, we have that D_{\sqsubseteq} is equivalent to S4 also when interpreted over the class of all linear orderings. Moreover, van Benthem considers also the case of D_{\sqsubseteq} interpreted over the integers, and proves that such logic is equivalent to the modal logic Grz, that is, S4 with the *Grzegorzczuk's Axiom*:

$$[D]([D](p \rightarrow [D]p) \rightarrow p) \rightarrow p,$$

expressing the fact that \sqsubseteq is well-founded. The satisfiability problem for both S4 and Grz is known to be PSPACE-complete [CR03, DG00, Lad77].

In [SS03], Shapirowsky and Shehtman explored the relations between the logic D_{\sqsupseteq} (where the strict subinterval relation \sqsupseteq is considered) and the logic of Minkowski space-time. The authors proved that the following axiomatic system is sound and complete for D_{\sqsupseteq} over the class of dense orderings:

- the K axiom;
- transitivity: $[D]p \rightarrow [D][D]p$,

- seriality: $\langle D \rangle \top$,
- 2-density: $\langle D \rangle p_1 \wedge \langle D \rangle p_2 \rightarrow \langle D \rangle (\langle D \rangle p_1 \wedge \langle D \rangle p_2)$.

By means of a suitable filtration technique, they also proved decidability and PSPACE completeness of this particular subinterval logic [Sha05].

We are not aware of published results about axiomatization and/or decidability of the logic of subintervals when the non-strict semantics and/or the proper subinterval relation \sqsubset is considered, over any class of linear orders.

1.3.5 The logics $\overline{B\overline{B}}$ and $\overline{E\overline{E}}$

The logics $\overline{B\overline{B}}$ and $\overline{E\overline{E}}$ are the fragments of HS featuring the modal operators $\langle B \rangle / \langle \overline{B} \rangle$ and $\langle E \rangle / \langle \overline{E} \rangle$, respectively. The syntax of $\overline{B\overline{B}}$ is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle B \rangle \varphi \mid \langle \overline{B} \rangle \varphi.$$

The syntax of $\overline{E\overline{E}}$ is defined by replacing $\langle B \rangle$ with $\langle E \rangle$ and $\langle \overline{B} \rangle$ with $\langle \overline{E} \rangle$, respectively.

The $\overline{B\overline{B}}$ and $\overline{E\overline{E}}$ logics are two example of how decidability can be achieved by making a suitable choice of the interval modalities. As shown by Goranko et al. [GMS04], the decidability of $\overline{B\overline{B}}$ and $\overline{E\overline{E}}$ can be obtained by embedding them into the propositional temporal logic of linear time $\text{TL}[F,P]$ with temporal modalities F (sometime in the future) and P (sometime in the past). The formulae of $\overline{B\overline{B}}$ are simply translated into formulae of $\text{TL}[F,P]$ by a mapping that replaces $\langle B \rangle$ by P and $\langle \overline{B} \rangle$ by F . $\text{TL}[F,P]$ has the finite model property and is decidable. $\overline{E\overline{E}}$ can be translated into $\text{TL}[F,P]$ in a similar way.

1.3.6 Split Logics

Split Logics (SLs for short) can be viewed as an attempt to identify expressive, yet decidable, propositional interval logics without imposing any locality principle. In the case of SLs, decidability is achieved by restricting the interval structures over which formulae are interpreted. In the following, we briefly describe such a semantic restriction, we outline the basic features of SLs, and we provide a short summary of the decidability results about them.

SLs have been proposed by Montanari, Sciavicco, and Vitacolonna in [MSV02]. They 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, and that at most two of its sub-intervals are ‘available’.

Formally, a split structure is a pair $\langle \mathbb{D}, \mathbb{H}(\mathbb{D}) \rangle$, where $\mathbb{H}(\mathbb{D})$ is a proper subset of $\mathbb{I}(\mathbb{D})$ (a precise characterization of $\mathbb{H}(\mathbb{D})$ can be found in [MSV02]). As an example, consider the split structure over the naturals $\langle \mathbb{N}, \text{SPLIT}(\mathbb{N}) \rangle$, where $\text{SPLIT}(\mathbb{N}) = \{[2^i a, 2^i(a+1)] : i, a \in \mathbb{N}\}$. It can be viewed as an upward unbounded layered structure, where the base layer (Layer 0) contains all and only the *atomic* intervals $[a, a+1] \in \text{SPLIT}(\mathbb{N})$ (an interval is atomic when it cannot be chopped into smaller subintervals), and the i -th layer contains all and only the intervals $[2^i a, 2^i(a+1)] \in$

SPLIT(\mathbb{N}). The relations between the split structure of the naturals and the corresponding upward unbounded layered structure is depicted in Figure 1.3. Arrows represent the ‘chopping’ relation between intervals.

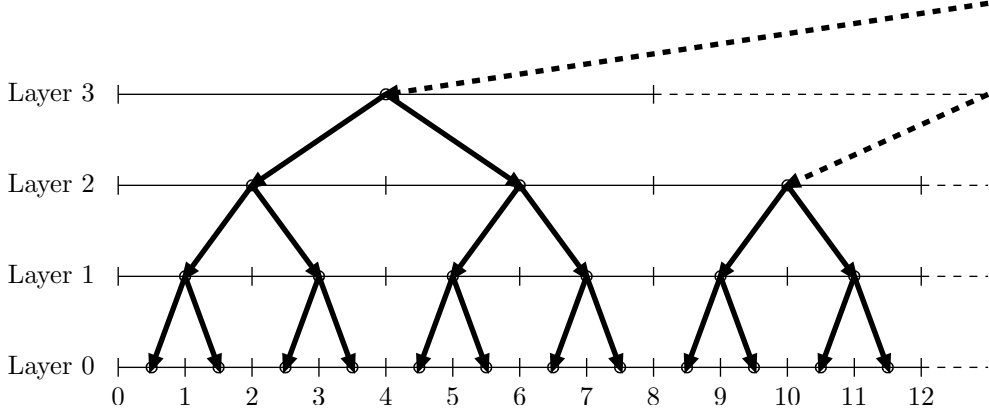


Figure 1.3: The split structure of the naturals.

Formulae of SLs are generated by the following abstract syntax:

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

Given a split structure $\langle \mathbb{D}, \mathbb{H}(\mathbb{D}) \rangle$, a *split model* is a tuple $\langle \mathbb{D}, \mathbb{H}(\mathbb{D}), \mathcal{V} \rangle$, where $\mathcal{V} : AP \rightarrow 2^{\mathbb{H}(\mathbb{D})}$. The formal semantics for the modal operators $\langle D \rangle$, $\langle \overline{D} \rangle$, $\langle F \rangle$, and $\langle \overline{F} \rangle$ are the following (the semantics for the ‘chop’ operators C , D , and T are as previously defined).

- $\mathbf{M}, [d_i, d_j] \Vdash \langle D \rangle\varphi$ iff there exists $[d_k, d_m] \in \mathbb{H}(\mathbb{D})$ such that $[d_k, d_m] \sqsubset [d_i, d_j]$ and $\mathbf{M}, [d_k, d_m] \Vdash \varphi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle \overline{D} \rangle\varphi$ iff there exists $[d_k, d_m] \in \mathbb{H}(\mathbb{D})$ such that $[d_i, d_j] \sqsubset [d_k, d_m]$ and $\mathbf{M}, [d_k, d_m] \Vdash \varphi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle F \rangle\varphi$ iff there exists $[d_k, d_m] \in \mathbb{H}(\mathbb{D})$ such that $d_j < d_k$ and $\mathbf{M}, [d_k, d_m] \Vdash \varphi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle \overline{F} \rangle\varphi$ iff there exists $[d_k, d_m] \in \mathbb{H}(\mathbb{D})$ such that $d_m < d_i$ and $\mathbf{M}, [d_k, d_m] \Vdash \varphi$.

As for decidability results, in [MSV02] Montanari, Sciavicco and Vitacolonna show that SLs can be viewed as the interval logic counterparts of the monadic first-order (MFO) theories of time granularity. By embedding them into decidable MFO theories of time granularity, the authors prove the decidability of various SLs, as well as their completeness with respect to the guarded fragment of these theories.

1.4 Neighborhood Logics

The interval logics based on Allen's relation *meet* and its inverse *met by* are called *neighborhood logics*. First-order neighborhood logics were introduced and studied by Zhou and Hansen in [CH98], while their propositional variants, interpreted over linear structures (both strict and non-strict), were first studied by Goranko, Montanari and Sciavicco in [GMS03a].

1.4.1 Propositional Neighborhood Logic

There are various choices of the language of Propositional Neighborhood Logic (PNL for short), depending on the choice for the temporal operators and on the considered semantics (strict and non-strict). Its most general version is denoted $\text{PNL}^{\pi+}$ and it includes the modal operators \diamond_r (*met by*) and \diamond_l (*meets*), and the modal constant π . Its formulae are generated by the following abstract syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \pi \mid \diamond_r\varphi \mid \diamond_l\varphi.$$

The language where the modal constant π is not included will be denoted PNL^+ . To make it easier to distinguish between the strict and the non-strict semantics, we will reserve the above notation for the case of non-strict PNL, while for the strict one, denoted PNL^- , we use $\langle A \rangle$ and $\langle \bar{A} \rangle$ instead of \diamond_r and \diamond_l , respectively (π is not included in PNL^-).

The formal semantics of the modal constant π is defined as in CDT, while for the non-strict operators \diamond_r and \diamond_l it is defined as follows:

- $\mathbf{M}^+, [d_i, d_j] \Vdash \diamond_r\psi$ iff $\exists d_k \in D, d_k \geq d_j$, such that $\mathbf{M}^+, [d_j, d_k] \Vdash \psi$;
- $\mathbf{M}^+, [d_i, d_j] \Vdash \diamond_l\psi$ iff $\exists d_k \in D, d_k \leq d_i$, such that $\mathbf{M}^+, [d_k, d_i] \Vdash \psi$.

In the case of the non-strict operators, the formal semantics is the following:

- $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A \rangle\psi$ iff $\exists d_k \in D, d_k > d_j$, such that $\mathbf{M}^-, [d_j, d_k] \Vdash \psi$;
- $\mathbf{M}^-, [d_i, d_j] \Vdash \langle \bar{A} \rangle\psi$ iff $\exists d_k \in D, d_k < d_i$, such that $\mathbf{M}^-, [d_k, d_i] \Vdash \psi$.

The non-strict operators $\langle A \rangle$ and $\langle \bar{A} \rangle$ can be defined in $\text{PNL}^{\pi+}$ as follows:

$$\langle A \rangle\varphi ::= \diamond_r(\neg\pi \wedge \varphi) \quad \langle \bar{A} \rangle\varphi ::= \diamond_l(\neg\pi \wedge \varphi)$$

It will turn out that the logic $\text{PNL}^{\pi+}$ subsumes both PNL^+ and PNL^- . The precise formalization and the proof of this result is given in Chapter 6, where the expressive power of $\text{PNL}^{\pi+}$ is compared with the one of the other propositional neighborhood logics.

Propositional Neighborhood Logic are quite expressive. Indeed, in the strict semantics various classes of linear structures can be characterized [GMS03b]:

(**A-SPNL^u**) $[A]p \rightarrow \langle A \rangle p$, in conjunction with its inverse, defines the class of *unbounded* structures;

(**A-SPNL^{de}**) $(\langle A \rangle \langle A \rangle p \rightarrow \langle A \rangle \langle A \rangle \langle A \rangle p) \wedge (\langle A \rangle [A]p \rightarrow \langle A \rangle \langle A \rangle [A]p)$, together with its inverse, defines the class of *dense* structures;

(**A-SPNL^{di}**) $([A] \perp \rightarrow [\bar{A}][A][A] \perp \vee \langle A \rangle (\langle A \rangle \top \wedge [A][A] \perp)) \wedge ((\langle A \rangle \top \wedge [A](p \wedge [\bar{A}]\neg p \wedge [A]p)) \rightarrow [\bar{A}][\bar{A}]\langle A \rangle (\langle A \rangle \neg p \wedge [A][A]p))$, with its inverse, defines the class of *discrete* structures;

(**A-SPNL^c**) $(\langle A \rangle \langle A \rangle [\bar{A}]p \wedge \langle A \rangle [A]\neg[\bar{A}]p) \rightarrow \langle A \rangle (\langle A \rangle [\bar{A}][\bar{A}]p \wedge [A]\langle A \rangle \neg[\bar{A}]p)$, with its inverse, defines the class of *Dedekind complete* structures.

Since PNL^- can be encoded into $\text{PNL}^{\pi+}$, we have that the above classes of structures can be defined in $\text{PNL}^{\pi+}$ as well.

As for axiomatizability results, in [GMS03b], Goranko et al. proposes sound and complete axiomatic systems for $\text{PNL}^{\pi+}$, PNL^+ , and PNL^- interpreted over the class of all linear orders. By combining such axiomatic systems with the above formulae, sound and complete axiomatic systems for $\text{PNL}^{\pi+}$ and PNL^- over the class of unbounded, dense, discrete, and Dedekind complete structures (and all their combinations) can be obtained.

1.4.2 Right Propositional Neighborhood Logic

In Chapter 3, we study a propositional interval temporal logic based on the right neighborhood relation between intervals, that we call Right Propositional Neighborhood Logic (RPNL for short). In its most general variant, called $\text{RPNL}^{\pi+}$, it is interpreted over non-strict interval structures and its formulae are recursively defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \pi \mid \diamond_r \varphi.$$

As in the case of PNL, we denote with RPNL^+ the fragment without the modal constant π , and with RPNL^- the fragment interpreted over strict interval structures, where \diamond_r is substituted with $\langle A \rangle$. The semantics of the temporal operators is the same as the one of PNL.

1.4.3 Branching Time Neighborhood Logic

All logics described so far are usually interpreted over linear structures, and feature temporal operators that only allow one to express properties of a single timeline. In Chapter 4, we discuss the decidability of an original propositional interval-based temporal logic, interpreted over infinite trees, that we call Branching Time Neighborhood Logic (BTNL for short). Such a logic combines the interval modalities of RPNL with the path quantifiers A and E of branching time temporal logics [Eme90]. To the best of our knowledge, BTNL is the first temporal logic that extends an interval temporal

logic with path quantifiers and that does not impose any semantic restriction (such as locality or homogeneity) on the valuation of formulae.

Formulae of BTNL are built from a set AP of propositional letters p, q, \dots , by using the Boolean connectives \neg and \vee , the modal constant π , and the future temporal operators $E\Diamond_r$ and $E\Box_r$. The other classical propositional connectives, as well as the logical constants \top (true) and \perp (false), are defined in the usual way. Furthermore, we introduce the temporal operator $A\Box_r$ as a shorthand for $\neg E\Diamond_r \neg$ and the temporal operator $A\Diamond_r$ as a shorthand for $\neg E\Box_r \neg$.

The formal syntax of BTNL is recursively defined by the following grammar:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \pi \mid E\Diamond_r\varphi \mid E\Box_r\varphi.$$

A formula of the form $E\Diamond_r\psi$, $E\Box_r\psi$, $A\Diamond_r\psi$ or $A\Box_r\psi$, is called a *temporal formula*. A temporal formula whose main temporal operator is either $E\Diamond_r$ or $E\Box_r$ is called an *existential formula*, while a temporal formula whose main temporal operator is either $A\Diamond_r$ or $A\Box_r$ is called a *universal formula*.

A *model* for a formula is a tuple $\mathbf{M}^+ = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+, \mathcal{V} \rangle$, where \mathbb{T} is a tree and $\mathcal{V} : AP \rightarrow 2^{\mathbb{I}(\mathbb{D})^+}$ is the *valuation function* that assigns to every propositional letter p the set of intervals where p holds. The semantics of BTNL is defined recursively by the *satisfiability relation* \Vdash as follows. Let $\mathbf{M}^+ = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+, \mathcal{V} \rangle$ be some given model, and let $[t_i, t_j] \in \mathbb{I}(\mathbb{D})^+$:

- $\mathbf{M}^+, [t_i, t_j] \Vdash \pi$ iff $t_i = t_j$;
- $\mathbf{M}^+, [t_i, t_j] \Vdash E\Diamond_r\psi$ iff there exists $t_k \in D$, $t_j \leq t_k$, such that $\mathbf{M}^+, [t_j, t_k] \Vdash \psi$;
- $\mathbf{M}^+, [t_i, t_j] \Vdash E\Box_r\psi$ iff there exists an infinite path $\rho = t_j t_{j+1} \dots$ rooted at t_j such that, for every t_k in π , with $t_j \leq t_k$, $\mathbf{M}^+, [t_j, t_k] \Vdash \psi$.

Putting together the tableau method for CTL [Eme90] and the one we developed for RPNL (Chapter 3), we build a doubly-exponential tableau-based decision procedure for BTNL, that is discussed in Chapter 4.

2

Tableaux and Dual-tableaux for Temporal Logics

In this chapter we first survey the main existing tableau methods for propositional point-based and interval-based temporal logics. Then we briefly describe the dual-tableau proof systems for standard relational logic, that will be used in Chapter 7 as the starting point for the development of a general proof system for interval temporal logics.

According to a common accepted perspective, tableau methods for modal and temporal logics can be classified as *explicit* or *implicit* [DGHP99]. Explicit methods keep track of the accessibility relation by means of some external device. One example is to maintain an auxiliary graph of named nodes n_i, n_j, \dots , where each node contains a subformula, or a set of subformulae, of the formula to be checked. The existence of an edge connecting n_i to n_j means that n_j is accessible from n_i . Another example is to associate structured labels to the nodes that constraint the formula, or the set of formulae, to hold only at the domain element(s) identified by the label. In this case, the accessibility relation is captured by the tableau system by means of labeled formulae. In implicit methods (c.f. [Fit83, Rau83]), the accessibility relation is built-in into the structure of the tableau. In this case, the tableau represents a model of the considered formula. The non-standard finite model property can then be exploited to show that the considered tableau method is an effective decision procedure (it does not lead to infinite computations). Another further classification is to partition implicit methods into *declarative* and *incremental* ones [KMMP93]. Methods in the former class first generate all possible sets of subformulae of a given formula, and then eliminate some (possibly all) of them, while those in the latter generate only ‘meaningful’ sets of formulae.

Relational logics provide a common background for a large class of relational structures used in computer science and can be used as a general framework for specification and reasoning in nonclassical logics. They are logical languages that describe how objects relates to other objects. In such logics, a set of primitive relations take the place of propositional variables and of predicate symbols, while the operations of relational algebra (e.g., negation, union, composition of relations) take the place of boolean connectives, quantifiers, and modalities. Dual-tableau proof systems for

relational logics were first developed in [Orl88] and further expanded in [GPO06a, Orl96]. The systems are founded on the Rasiowa-Sikorski system for the first order logic [RS63] which is extended with the rules for equality predicate in [GPO06b].

2.1 Tableaux for point-based temporal logics

The problem of devising tableau methods for propositional point-based temporal logics, such as Linear Temporal Logic (LTL) and Computational Tree Logic (CTL) (and various fragments and extensions of them), has been extensively investigated in the literature. In this section we briefly discuss the tableau for LTL originally proposed by Wolper in [Wol85], its incremental improvement (Kesten et al., [KMMP93]), and the implicit tableau for CTL of Emerson and Halpern [EH85].

2.1.1 Tableaux for Linear Temporal Logic

LTL extends propositional logic by adding a unary modality X (*next*) and a binary modality U (*until*). Past LTL (PLTL) extends LTL with the past modalities \bar{X} (*previous*) and S (*since*). These logics are usually interpreted over infinite sequences of states, namely, over linear structures isomorphic to the set of natural/integer numbers, with the usual ordering relation. Let us consider, for simplicity, the case of LTL. A model M for LTL is a pair $\langle S, \mathcal{V} \rangle$, where S is a state sequence d_0, d_1, d_2, \dots , and \mathcal{V} is a valuation function that associates to every propositional variable p the set of states where p holds. We have that $X\varphi$ holds at a state d_i if and only if φ holds at d_{i+1} , and $\varphi U \psi$ holds at d_i if and only if there exist a state d_j in the future of d_i such that ψ holds at d_j and φ holds at every state between d_i and d_j (excluding d_j). The satisfiability problem for both LTL and PLTL is PSPACE-complete [SC85].

In the following, we first describe the exponential time declarative method for LTL developed by Wolper [Wol85] and successively extended by Lichtenstein and Pnueli to PLTL [LP00]. Then we discuss the improved incremental method for PLTL proposed by Kesten et al. [KMMP93].

Wolper's tableau method is a natural extension of the one for propositional logic. In the classical setting, the formula to check for satisfiability is turned into a tree-like structure, annotated with its subformulae, to take into account the different possibilities that come from disjunctions. In the temporal setting, the situation is a bit more complex because formulae are interpreted over sequences of states. The key idea of the method is the so-called *fixpoint definition* of temporal operators, that allows one to split every formula into a (possibly empty) part related to the current state and a part related to the next (resp. previous) state. For example, the formula $\varphi U \psi$ is split as follows: either ψ holds now, or φ holds now and $\varphi U \psi$ holds at the next state. The fact that only a finite set of different scenarios can be generated in this way allows one to devise a mechanism to identify periodic situations in a finite time.

Let φ be the PLTL-formula to check for satisfiability. The set of all subformulae of φ and their negations is called the *closure* of φ , and it is denoted by $\text{CL}(\varphi)$. The key notion of this method is the notion of *atom*, namely, any subset A of $\text{CL}(\varphi)$ such that:

1. for all $\psi \in \text{CL}(\varphi)$, $\psi \in A$ if and only if $\neg\psi \notin A$;
2. for all $\psi \vee \xi \in \text{CL}(\varphi)$, $\psi \vee \xi \in A$ if and only if either $\psi \in A$, or $\xi \in A$;
3. for all $\psi U \xi \in \text{CL}(\varphi)$ (similarly for $\psi S \xi$), if $\psi U \xi \in A$ then $\psi \in A$ or $\xi \in A$, and if $\xi \in A$ then $\psi U \xi \in A$.

Intuitively, an atom is a consistent subset of $\text{CL}(\varphi)$. Wolper's method builds a directed graph $G = \langle N, E \rangle$, whose nodes are all the atoms and whose edges are such that $(n_i, n_j) \in E$ if and only if:

1. for all $X\psi \in \text{CL}(\varphi)$ (similarly for $\bar{X}\psi$), $X\psi \in n_i$ if and only if $\psi \in n_j$;
2. for all $\psi U \xi \in \text{CL}(\varphi)$ (similarly for $\psi S \xi$), if $\psi U \xi \in n_i$ and $\neg\xi \in n_i$, then $\psi U \xi \in n_j$, and if $\psi U \xi \in n_j$ and $\psi \in n_i$, then $\psi U \xi \in n_i$.

A node n_i that does not contain any formula $\bar{X}\psi \in \text{CL}(\varphi)$ and such that, for every $\psi S \xi \in n_i$, $\xi \in n_i$, is called an *initial node*.

The procedure attempts to obtain a model for the formula by searching for a suitable infinite path n_0, n_1, \dots starting from an initial node n_0 that contains φ . To this end, it exploits the notion of *fulfilling path*. An infinite path n_0, n_1, \dots is a fulfilling path if and only if for every $i \geq 0$, if $\psi U \xi \in n_i$, then there exists $j \geq i$ such that $\xi \in n_j$. We have that φ is satisfiable if and only if there exists a fulfilling path for φ in G .

Since fulfilling paths are infinite objects, we need a finite characterization of them. This is given in terms of self-fulfilling maximal strongly connected components of G . A strongly connected component C is *self-fulfilling* if for every formula $\psi U \xi$ that belongs to a node $n_i \in C$ there exists a node $n_j \in C$ such that $\xi \in n_j$. The algorithm that checks for φ satisfiability works as follows: first (*construction phase*), it builds the graph G ; then (*elimination phase*), it removes maximal strongly connected components that are either not reachable from an initial node, or are without outgoing edges and not self-fulfilling. It turns out that φ is satisfiable if and only if the final graph G is not empty.

An efficient incremental variant of Wolper's declarative procedure is the tableau method of Kesten et al. [KMMP93]. This method extends to PLTL the incremental method for LTL originally developed by Pnueli and Sherman [PS81]. Like Wolper's method, it is based on the notion of atom. However, instead of building the entire set of atoms immediately (thus paying the worst case exponential complexity price), it builds the tableau incrementally, introducing only those atoms that are necessary to decide the satisfiability of the given formula. The construction starts from an initial graph that contains all initial atoms including φ , all connected to an empty node n_0 (*initial phase*). Then (*correct-graph phase*), as long as some edge (n_i, n_j) in the current graph G violates some future (resp. past) constraints, that is, $X\psi \in n_i$ and

$\psi \notin n_j$ (resp. $\bar{X}\psi \in n_j$ and $\psi \notin n_i$), it adds a possibly new atom n_k that contains all formulae of n_j and some additional formulae that remove the violations. Furthermore, it replaces the unsatisfactory edge (n_i, n_j) by a new edge (n_i, n_k) (resp. (n_k, n_i)).

Once all violations has been removed, the algorithm searches for a self-fulfilling maximal strongly connected component that is reachable from an initial node containing φ . If such a component exists, the formula is satisfiable, otherwise not. Even though in the worst case the computational complexity of this procedure is still exponential, in many cases a much smaller number of atoms is generated.

2.1.2 Tableaux for Computational Tree Logic

A propositional temporal logic interpreted over branching structures for which a tableau method has been devised is CTL. CTL extends propositional logic with two unary temporal operators EX , AX and two binary temporal operators EU and AU . A model for CTL is a tuple $\mathbf{M} = \langle S, R, \mathcal{V} \rangle$ where S is a set of states, R is a total binary relation, and \mathcal{V} is the valuation function. The unfolding of \mathbf{M} produces a tree, whose paths are isomorphic to the set of naturals. The formula $EX\varphi$ (resp., $AX\varphi$) states that φ holds at some (resp., every) successor of the current state, while the formula $EU(\varphi, \psi)$ states that along some (resp., every) path starting from the current state, the formula φ holds until ψ holds. The satisfiability problem for CTL is EXPTIME-complete.

An implicit tableau method for CTL, that generalizes the one for LTL, has been proposed by Emerson and Halpern in [EH85]. If φ is the formula to be checked, the procedure generates a directed graph $G = \langle N, E \rangle$ whose nodes are maximal propositionally consistent subsets of $\text{CL}(\varphi)$. Furthermore, it puts an edge from n_i to n_j if and only if the following conditions hold:

1. for all $AX\psi \in \text{CL}(\varphi)$, if $AX\psi \in n_i$ then $\psi \in n_j$ (the same for $\neg EX\psi$);
2. for all $AU(\psi, \xi) \in \text{CL}(\varphi)$, if $AU(\psi, \xi) \in n_i$ then either $\xi \in n_i$ or $\psi \in n_i$ and $AU(\psi, \xi) \in n_j$.

As in the case of Wolper's tableau, after the initial construction phase there is an elimination phase. In this case it encompass both a local pruning that removes local inconsistencies and another pruning process that removes nodes including eventualities that are not fulfilled in the current graph. The formula is satisfiable if and only if the final graph is not empty.

2.2 Tableaux for interval-based temporal logics and Duration Calculi

In the literature there exist very few tableau methods for interval-based temporal logics and duration calculi. Here we briefly survey the tableau decision procedure for an extension of Local PITL interpreted over finite state sequences (LPITL_{proj}) proposed by Bowman and Thompson in [BT03], the one for a fragment of Propositional

Duration Calculus (PDC_{pos}) proposed by Chetcuti-Serandio and Fariñas del Cerro in [CSFDC00], and the semi-decision tableau method for CDT interpreted over partial orders (BCDT⁺) by Goranko et al. [GMS03a, GMSS06].

2.2.1 A tableau for $LPITL_{proj}$

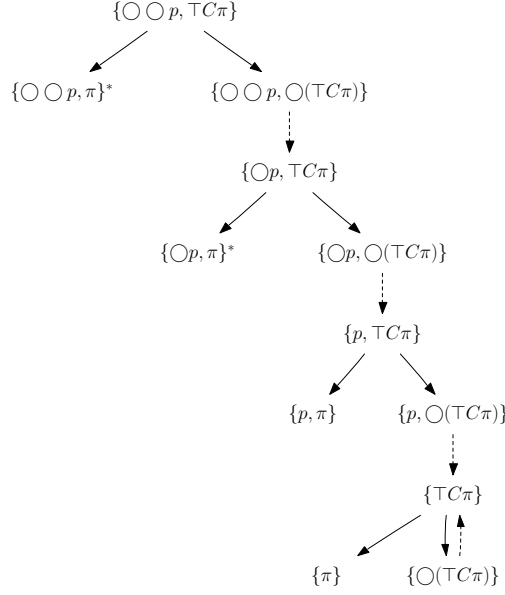
$LPITL_{proj}$ pairs the operators \bigcirc (*strong next*) and C (*chop*) of $LPITL$ with a new binary operator $proj$ (*projection*) that represents repetitive behaviors. The formula $\bigcirc\varphi$ holds on the current interval if and only if φ holds over an interval of length one less than the current interval, resulting from moving one state into the future. $\varphi C \psi$ holds on the current interval if it can be partitioned into two sub-intervals, the first of which satisfies φ and the second of which satisfies ψ . For any given pair of formulae φ and ψ , $\varphi proj \psi$ holds over an interval if it can be partitioned into a series of sub-intervals each of which satisfies φ , while ψ (the *projected formula*) holds on the interval formed by the end points of these sub-intervals. $LPITL_{proj}$ formulae are interpreted over finite state sequences d_0, d_1, \dots, d_k . The locality constraint forces that, for any propositional variable p and any interval $[d_i, d_j]$, $[d_i, d_j] \in \mathcal{V}(p)$ if and only if $[d_i, d_i] \in \mathcal{V}(p)$. The satisfiability problem for $LPITL_{proj}$ is non-elementary [GMS04].

Bowman and Thompson's tableau method is based on the definition of a suitable normal form for all operators of the logic, which reflects the locality constraint and provides uniform inductive definitions of the operators. Starting from them, Bowman and Thompson develop an implicit tableau-based decision procedure for satisfiability checking in the style of Wolper [Wol85]. The normal form for $LPITL_{proj}$ formulae has the following general form:

$$(\pi \wedge \varphi_e) \vee \bigvee_i (\varphi_i \wedge \bigcirc\varphi'_i),$$

where π stands for the formula $\bigcirc\perp$ that holds only on point-intervals, φ_e and φ_i are point formulae, and φ'_i is an arbitrary $LPITL_{proj}$ formula. The first disjunct states when a formula is satisfied on a point-interval, while the second disjunct states the possible ways in which a formula can be satisfied over a strict interval, namely, a point formula must hold over the initial point and an arbitrary formula must hold over the remainder of the interval.

The tableau construction exploits this normal form to split the requirements imposed by a temporal formula into requirements about the present (the initial point of the interval) and requirements into the future (the remainder of the interval). As in the case of Wolper's tableau, it generates a directed graph $G = \langle N, E \rangle$, where each node corresponds to a state of the model and is labeled by a set of formulae. Given a formula φ to test for satisfiability, the construction of G starts from the initial node n_0 labeled with the set $\{\varphi, \top C \pi\}$. The expansion rules for the Boolean connectives are the standard ones, while formulae of the forms $\psi C \xi$ and $\psi proj \xi$ (as well as $\neg(\psi C \xi)$ and $\neg(\psi proj \xi)$) are expanded by exploiting the normal forms of their subformulae. Finally, formulae of the form $\bigcirc\psi$ are expanded by adding a new node, corresponding to a new state, labeled with ψ .

Figure 2.1: An example of the tableau method for LPITL_{proj}

Once that the construction of G has been terminated, the procedure looks for unsatisfiable nodes in G and marks them. A node is unsatisfiable if one of the following conditions hold:

1. it contains a formula and its negation;
2. it contains both a formula $\circ\psi$ and π ;
3. all of its successors are unsatisfiable.

It turns out that φ is satisfiable if and only if the initial node is not marked. Figure 2.1 depicts the tableau for the satisfiable formula $\circ\circ p$. Dashed arrows represent the expansion of $\circ\psi$ formulae, while marked nodes are identified by an asterisk (*).

2.2.2 A tableau for Propositional Duration Calculus

In [CSFDC00], Chetcuti-Serandio e Fariñas del Cerro isolate a fragment of Propositional Duration Calculus, called PDC_{pos} , which includes the operators \wedge , \vee and C , but not \neg . PDC_{pos} is expressive enough to capture Allen's relations [All83] and decidable. The language is also characterized by a special constant, that is, the constant l , whose interpretation can vary over time, denoting *the length of the current interval*. It is combined with the structure of the additive group of the reals as temporal domain, which allows computing lengths of concatenated intervals, and so on. Another specific feature of Duration Calculus, that is preserved in PDC_{pos} , is the special category of

terms called *state expressions* which are used to represent the duration for which a system stays in a particular state.

Tableau nodes are conjunctions of labeled formulae, labeled state expressions, and constraints. Labeled formulae (resp., labeled state expressions) are pairs $\langle \varphi, [d_i, d_j] \rangle$ (resp., $\langle \sigma, [d_i, d_j] \rangle$), where φ (resp., σ) is a formula (resp., state expression) and $[d_i, d_j]$ is an interval. Constraints can be either *qualitative*, e.g., $d_i \leq d_j$, and *quantitative*, e.g., $d_j - d_i = k$ or $d_j - d_i > k$, where k is a constant.

The tableau construction starts from an initial node including the labeled formula $\langle \varphi, [d_0, d_1] \rangle$, where φ is the formula to be checked and $[d_0, d_1]$ is a generic interval, and it proceeds by applying suitable expansion rules to labeled formulae or labeled state expressions in the leaf of the considered branch. Closing rules detect contradictory formulae associated to the same interval, or sets of inconsistent constraints. The proof of termination exploits the fact that each expansion rule can be applied only finitely often to any branch, while the soundness and completeness proof exploits the fact that the expansion rules preserve a suitable notion of satisfiability. Complexity issues are not addressed.

2.2.3 A tableau for BCDT⁺

The last tableau method we describe in this chapter is the one for BCDT⁺ proposed by Goranko et al. [GMS03a, GMSS06]. As shown in Chapter 1, BCDT⁺ is a generalization of Venema's CDT logic to (non-strict) partial orderings with the linear interval property, and it is undecidable. Thus, the proposed tableau method is not guaranteed to be terminating, and it is only a semi-decision procedure. However, it can easily be adapted to variations and subsystems of BCDT⁺, providing a general tableau method for propositional interval temporal logics.

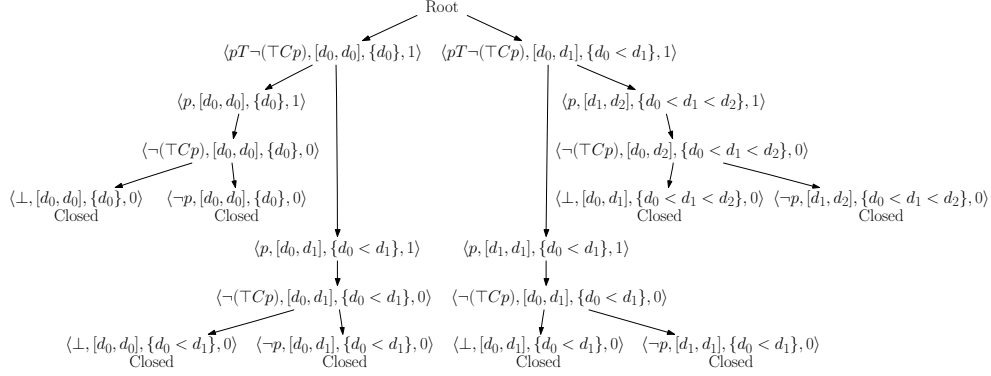
The tableau construction generates a (possibly infinite) tree, whose nodes are decorated with labeled formulae $\langle \psi, [d_i, d_j], \mathbb{D}, u \rangle$, where $\mathbb{D} = \langle D, < \rangle$ is a finite partial order with the linear interval property, $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, and u is a *local flag function* which associates the values 0 or 1 with every branch B containing the node. Intuitively, the value 0 for a node n in a branch B means that n can be expanded on B . If B is a branch, then $B \cdot n$ is the result of expanding B with the node n , while $B \cdot n_1 | \dots | n_k$ is the result of expanding B with k immediate successor nodes n_1, \dots, n_k . With \mathbb{D}_B we denote the finite partial ordering in the leaf of B .

The construction of a tableau for BCDT⁺ starts from a three-node *initial tree* built up from an empty-decorated root and two leaves with decorations $\langle \varphi, [d_0, d_0], \{d_0\}, 0 \rangle$ and $\langle \varphi, [d_0, d_1], \{d_0 < d_1\}, 0 \rangle$, where φ is the formula to test for satisfiability. The procedure exploits the following expansion rules to add new nodes to the tree.

Definition 2.1. Given a tree \mathcal{T} , a branch B in \mathcal{T} , and a node $n \in B$ decorated with $\langle \psi, [d_i, d_j], \mathbb{D}, u_n \rangle$ such that $u_n(B) = 0$, the *branch expansion rule* for B and n is defined as follows. In all considered cases, $u_{n'}(B') = 0$ for all new nodes n' and branches B' .

- If $\psi = \neg \xi$, expand B to $B \cdot n_0$, where n_0 is decorated with $\langle \xi, [d_i, d_j], \mathbb{D}_B, u_{n_0} \rangle$.

- If $\psi = \xi_0 \vee \xi_1$, then expand B to $B \cdot n_0 | n_1$, where n_0 is decorated with $\langle \xi_0, [d_i, d_j], \mathbb{D}_B, u_{n_0} \rangle$ and n_1 is decorated with $\langle \xi_1, [d_i, d_j], \mathbb{D}_B, u_{n_1} \rangle$.
- If $\psi = \neg(\xi_0 \vee \xi_1)$, then expand B to $B \cdot n_0 \cdot n_1$, where n_0 is decorated with $\langle \neg\xi_0, [d_i, d_j], \mathbb{D}_B, u_{n_0} \rangle$ and n_1 is decorated with $\langle \neg\xi_1, [d_i, d_j], \mathbb{D}_B, u_{n_1} \rangle$.
- If $\psi = \neg(\xi_0 C \xi_1)$ and d is an element of \mathbb{D}_B such that $d_i \leq d \leq d_j$ and d has not been used yet to expand n in B , then expand B to $B \cdot n_0 | n_1$, where n_0 is decorated with $\langle \neg\xi_0, [d_i, d], \mathbb{D}_B, u_{n_0} \rangle$ and n_1 is decorated with $\langle \neg\xi_1, [d, d_j], \mathbb{D}_B, u_{n_1} \rangle$.
- If $\psi = \neg(\xi_0 D \xi_1)$ and d is an element of \mathbb{D}_B such that $d \leq d_i$ and d has not been used yet to expand n in B , then expand B to $B \cdot n_0 | n_1$, where n_0 is decorated with $\langle \neg\xi_0, [d, d_i], \mathbb{D}_B, u_{n_0} \rangle$ and n_1 is decorated with $\langle \neg\xi_1, [d, d_j], \mathbb{D}_B, u_{n_1} \rangle$.
- If $\psi = \neg(\xi_0 T \xi_1)$ and d is an element of \mathbb{D}_B such that $d \geq d_j$ and d has not been used yet to expand n in B , then expand B to $B \cdot n_0 | n_1$, where n_0 is decorated with $\langle \neg\xi_0, [d_j, d], \mathbb{D}_B, u_{n_0} \rangle$ and n_1 is decorated with $\langle \neg\xi_1, [d_i, d], \mathbb{D}_B, u_{n_1} \rangle$.
- If $\psi = \xi_0 C \xi_1$, then expand the branch B to $B \cdot (n_i \cdot m_i) | \dots | (n_j \cdot m_j) | (n'_i \cdot m'_i) | \dots | (n'_{j-1} \cdot m'_{j-1})$, where:
 1. for all $d_i \leq d_k \leq d_j$, n_k is decorated with $\langle \xi_0, [d_i, d_k], \mathbb{D}_B, u_{n_k} \rangle$ and m_k is decorated with $\langle \xi_1, [d_k, d_j], \mathbb{D}_B, u_{m_k} \rangle$;
 2. for all $i \leq k \leq j-1$, \mathbb{D}_k is the partial ordering obtained by inserting a new element d between d_k and d_{k+1} , n'_k is decorated with $\langle \xi_0, [d_i, d], \mathbb{D}_k, u_{n'_k} \rangle$ and m'_k is decorated with $\langle \xi_1, [d, d_j], \mathbb{D}_k, u_{m'_k} \rangle$.
- If $\psi = \xi_0 D \xi_1$, then repeatedly expand the current branch, once for every element $d \leq d_i$, by adding the subtree $(n_0 \cdot m_0) | (n_1 \cdot m_1) | (n_2 \cdot m_2)$ to the leaf of B , where:
 1. n_0 is decorated with $\langle \xi_0, [d, d_i], \mathbb{D}_B, u_{n_0} \rangle$ and m_0 is decorated with $\langle \xi_1, [d, d_j], \mathbb{D}_B, u_{m_0} \rangle$;
 2. \mathbb{D}' is the partial ordering obtained by inserting a new element $d' < d$ which is incomparable with all existing predecessors of d , n_1 is decorated with $\langle \xi_0, [d', d_i], \mathbb{D}', u_{n_1} \rangle$, and m_1 is decorated with $\langle \xi_1, [d', d_j], \mathbb{D}', u_{m_1} \rangle$;
 3. if $d = d_i$, then do not add the subtree $(n_2 \cdot m_2)$ to B ;
 4. if $d < d_i$, \mathbb{D}' is the partial ordering obtained by inserting a new immediate successor d' of d in $[d, d_i]$, n_2 is decorated with $\langle \xi_0, [d', d_i], \mathbb{D}', u_{n_2} \rangle$, and m_2 is decorated with $\langle \xi_1, [d', d_j], \mathbb{D}', u_{m_2} \rangle$.
- If $\psi = \xi_0 T \xi_1$, then repeatedly expand the current branch, once for every element $d \geq d_j$, by adding the subtree $(n_0 \cdot m_0) | (n_1 \cdot m_1) | (n_2 \cdot m_2)$ to the leaf of B , where:
 1. n_0 is decorated with $\langle \xi_0, [d_j, d], \mathbb{D}_B, u_{n_0} \rangle$ and m_0 is decorated with $\langle \xi_1, [d_i, d], \mathbb{D}_B, u_{m_0} \rangle$;

Figure 2.2: An example of the tableau method for BCDT⁺

2. \mathbb{D}' is the partial ordering obtained by inserting a new element $d' > d$ which is incomparable with all existing successors of d , n_1 is decorated with $\langle \xi_0, [d_j, d'], \mathbb{D}', u_{n_1} \rangle$, and m_1 is decorated with $\langle \xi_1, [d_i, d'], \mathbb{D}', u_{m_1} \rangle$;
3. if $d = d_j$, then do not add the subtree $(n_2 \cdot m_2)$ to B ;
4. if $d > d_j$, \mathbb{D}' is the partial ordering obtained by inserting a new immediate predecessor d' of d in $[d_j, d]$, n_2 is decorated with $\langle \xi_0, [d_j, d'], \mathbb{D}', u_{n_2} \rangle$, and m_2 is decorated with $\langle \xi_1, [d_i, d'], \mathbb{D}', u_{m_2} \rangle$.

Finally, for every node $m \neq n$ in B and any branch B' extending B , let $u_m(B') = u_m(B)$, while for every branch B' extending B , $u_n(B') = 1$, unless $\psi = \neg(\xi_0 C \xi_1)$, $\psi = \neg(\xi_0 D \xi_1)$, or $\psi = \neg(\xi_0 T \xi_1)$ (in such cases $u_n(B') = 0$).

We briefly explain the expansion rules for $\xi_0 C \xi_1$ and $\neg(\xi_0 C \xi_1)$ (similar considerations can be made for the cases of the temporal operators D and T). The rule for the formula $\xi_0 C \xi_1$ deals with two possible cases: either there exists $d_k \in \mathbb{D}_B$ such that ξ_0 holds over $[d_i, d_k]$ and ξ_1 holds over $[d_k, d_j]$, or such an element must be added to \mathbb{D}_B . On the converse, the formula $\neg(\xi_0 C \xi_1)$ states that, for all $d_i \leq d \leq d_j$, ξ_0 does not hold over $[d_i, d]$ or ξ_1 does not hold over $[d, d_j]$. The expansion rule imposes such a condition for a single element d and keeps the flag equal to 0. In this way, all elements of \mathbb{D}_B are eventually considered, including those elements that will be added in some subsequent steps of the tableau construction.

To determine whether a branch can be further expanded or not, suitable notions of open and closed branch are defined. A branch is *closed* if one of the following conditions holds:

1. there are two nodes n, n' in B such that n is decorated with $\langle \psi, [d_i, d_j], \mathbb{D}, u_n \rangle$ and n' is decorated with $\langle \neg\psi, [d_i, d_j], \mathbb{D}', u_{n'} \rangle$, for some formula ψ ;
2. there is a node n decorated with $\langle \pi, [d_i, d_j], \mathbb{D}, u_n \rangle$ such that $d_i \neq d_j$;

3. there is a node n decorated with $\langle -\pi, [d_i, d_j], \mathbb{D}, u_n \rangle$ such that $d_i = d_j$.

If none of the above conditions hold, the branch is *open*. The expansion strategy for the tableau expands a branch B only if it is open and it applies the branch expansion rule to the closest to the root node for which the branch expansion rule is applicable. Figure 2.2 depicts a closed tableau for the unsatisfiable formula $p T \neg(\top C p)$.

It can be proved that this method is sound and complete. However, it is not guaranteed to be terminating: the expansion rules can be applied infinitely often, and the resulting tableau can be an infinite object.

2.3 Dual-tableaux for standard relational logic

In this section we describe a proof system in the style of dual tableau for the classical relational logic of binary relations, RL, which provides a means for proving the identities valid in the class of representable relation algebras [GPO06a, Or196]. The vocabulary of RL consists of a set $\mathbb{O}\mathbb{V} = \{x, y, z, \dots\}$ of *object variables*, a set $\mathbb{R}\mathbb{V} = \{R, S, \dots\}$ of *relational variables* and a set $\mathbb{O}\mathbb{P} = \{-, \cup, \cap, ;\}$ of *relational operation symbols*. Formulae of RL are of the form $x R y$, where R is a *relational term* built up by composing relational variables in $\mathbb{R}\mathbb{V}$ with the operators in $\mathbb{O}\mathbb{P}$.

The semantics of RL is given in terms of RL-models defined as pairs $\langle D, m \rangle$, where D is the domain over which RL-formulae are interpreted and m is a *meaning function* that assigns to every relational variable $R \in \mathbb{R}\mathbb{V}$ a binary relation in $D \times D$. The semantics of compound relational terms reflects the semantics of the operators in $\mathbb{O}\mathbb{P}$.

The dual-tableau proof system for RL we present in this section is the one originally developed in [Or188]. Such a proof system was subsequently expanded by including equality and the universal relation in the language [GPO06a, Or196]. It consists of axiomatic sets of formulae and rules which apply to finite sets of formulae. The axiomatic sets take the place of axioms. The rules have the following general form:

$$(*) \quad \frac{\Phi}{\Phi_1 \mid \dots \mid \Phi_n}$$

where Φ_1, \dots, Φ_n are finite non-empty sets of formulae, $n \geq 1$, and Φ is a finite (possibly empty) set of formulae. Φ is called the *premise* of the rule, and Φ_1, \dots, Φ_n are called its *conclusions*. A rule of the form $(*)$ is said to be *applicable* to a set X of formulae whenever $\Phi \subseteq X$. As a result of application of a rule of the form $(*)$ to a set X , we obtain the sets $(X \setminus \Phi) \cup \Phi_i$, $i = 1, \dots, n$. As usual, any concrete rule will always be presented in a short form without set brackets.

In dual tableau systems proofs have the form of finitely branching trees. Branching is interpreted as conjunction and the sets of formulae in the nodes of the trees are interpreted as disjunctions of their members. A branch of the proof tree is closed whenever it contains a node with an axiomatic set of formulae. A tree is closed if all of its branches are closed. A formula is provable whenever there exists a closed proof tree for it.

Let x, y, z be object variables and R, S relational terms. An *axiomatic set* is any set of relational formulae that contains both $x R y$ and $x -R y$. The rules for the standard relational logic RL are the following.

$$\begin{array}{ll}
(\cup) \quad \frac{x (R \cup S) y}{x R y, x S y} & (-\cup) \quad \frac{x -(R \cup S) y}{x -R y \mid x -S y} \\
(\cap) \quad \frac{x (R \cap S) y}{x R y \mid x S y} & (-\cap) \quad \frac{x -(R \cap S) y}{x -R y, x -S y} \\
(-) \quad \frac{x - -R y}{x R y} & \\
(-1) \quad \frac{x R^{-1} y}{y R x} & (-^{-1}) \quad \frac{x -R^{-1} y}{y -R x} \\
(;) \quad \frac{x (R; S) y}{x R z, x (R; S) y \mid z S y, x (R; S) y} & z \text{ is any variable} \\
(-;) \quad \frac{x -(R; S) y}{x -R z, z -S y} & z \text{ is a new variable}
\end{array}$$

In [Orl88], the author proves that such a proof system is sound and complete for RL. Soundness establishes that every formula with a closed proof tree is valid, while completeness states that any valid formula has a closed proof tree. The latter is proved by contradiction. She consider an open proof tree for the formula. It necessarily has an infinite branch (this is guaranteed by König's lemma). Then she makes this tree complete: whenever a rule is applicable to a node of the tree, then it has been applied. Next, from the syntactic resources of an infinite branch she can construct a suitable RL-model and she proves that it is a model that contradicts the formula. It is worth noticing that the standard relational logic RL is not decidable. Hence, the dual-tableau proof system presented here is not guaranteed to be terminating.

The applications of relational logics and dual-tableau proof systems to modal logics was first explored by Orłowska in [Orl88] and then formalized in the “formulas are relations” paradigm [Orl94]. Since then relational proof systems have been developed for several logical theories, like Hoare relations [DOR94], intuitionistic logics [FO95], finite-valued logics [KMO98], linear logics [Mac97], Lambek Calculus [Mac98, MO02], temporal logics [Orl95], and other modal logics [Orl96]. Building up from those approaches, in Chapter 7 we will devise an original relational proof system for interval temporal logics.

3

The tableau method for RPNL

This chapter is devoted to an optimal tableau-based procedure for the satisfiability problem for RPNL over natural numbers and/or finite linear orderings. The proposed tableau method partly resembles the tableau-based decision procedure for LTL [Wol85]. 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.

3.1 An intuitive account of the method

Before describing the tableau-based decision procedure for RPNL in details, we give an intuitive account of it by introducing 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.

For the sake of simplicity we consider the strict semantics of RPNL. Later on, we will show how to extend this process to the case of RPNL ^{π^+} . Let $D = \{d_0, d_1, d_2, \dots\}$ 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 the 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 $\text{REQ}(d_2)$ 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 $\text{REQ}(d_1)$ 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 $[A] \psi \in \text{REQ}(d_1)$, $\psi \in A_{[d_1, d_3]}$. 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 $[A] \psi \in \text{REQ}(d_2)$, $\psi \in A_{[d_2, d_3]}$. 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.

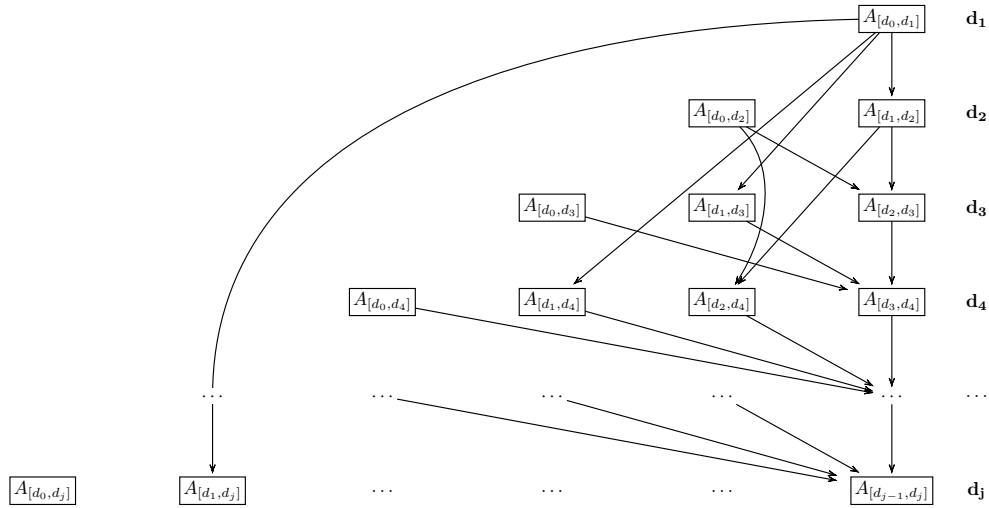


Figure 3.1: The layered structure

The layered structure generated by the process is graphically depicted in Figure 3.1. The first layer correspond 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 [BM05b], 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 3.1 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 [GMS03a, GMSS06], 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 that works for all variants of RPNL ($\text{RPNL}^{\pi+}$, RPNL^+ , and 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 3.1. In such a way, the procedure avoids the double exponential blow-up of the method given in [BM05b]. This chapter is a revised and extended version of [BMS07b].

3.2 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. From now on, we will consider the case of the logic RPNL^{π^+} (the cases of the logics RPNL^+ and RPNL^- are discussed in Section 3.5). Furthermore, we restrict our attention to non-strict interval structures $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$ where \mathbb{D} is isomorphic to the set of natural numbers (with the usual ordering) or to a prefix of them.

Let φ be an RPNL^{π^+} formula to be checked for satisfiability and let AP be the set of its propositional letters. Since RPNL has no past-time operators we say that φ is satisfiable if and only if it holds over an interval $[d_0, d_i]$, where d_0 is the least point of D and $d_0 \leq d_i$. In such a case, the *initial interval* $[d_0, d_0]$ is such that $\diamond_r \varphi$ holds on it. For the sake of brevity, we use $\circ_r \psi$ as a shorthand for both $\diamond_r \psi$ and $\square_r \psi$.

Definition 3.1. The *closure* $\text{CL}(\varphi)$ of φ is the set of all subformulae of $\diamond_r \varphi$ and of their negations (we identify $\neg\neg\psi$ with ψ).

Definition 3.2. The set of *temporal requests* of φ is the set $\text{TF}(\varphi)$ of all temporal formulae in $\text{CL}(\varphi)$, that is, $\text{TF}(\varphi) = \{\circ_r \psi \in \text{CL}(\varphi)\}$.

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

Proposition 3.3. *For every formula φ , $|\text{CL}(\varphi)|$ is less than or equal to $2 \cdot (|\varphi| + 1)$, while $|\text{TF}(\varphi)|$ is less than or equal to $2 \cdot |\varphi|$.*

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

Definition 3.4. A φ -atom is a set $A \subseteq \text{CL}(\varphi)$ such that:

- for every $\psi \in \text{CL}(\varphi)$, $\psi \in A$ iff $\neg\psi \notin A$;
- for every $\psi_1 \vee \psi_2 \in \text{CL}(\varphi)$, $\psi_1 \vee \psi_2 \in A$ iff $\psi_1 \in A$ or $\psi_2 \in A$.

We denote the set of all φ -atoms by A_φ . We have that $|A_\varphi| \leq 2^{|\varphi|+1}$. Atoms are connected by the following binary relation.

Definition 3.5. 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 $\square_r \psi \in \text{CL}(\varphi)$, if $\square_r \psi \in A$, then $\psi \in A'$.

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

Definition 3.6. A (non-strict) φ -labelled interval structure (LIS for short) is a tuple $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$ is a non-strict interval structure and $\mathcal{L} : \mathbb{I}(\mathbb{D})^+ \rightarrow A_\varphi$ is a *labelling function* such that (a) for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, $\pi \in \mathcal{L}([d_i, d_j])$ iff $d_i = d_j$, and (b) 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, that of the modal constant π , and that of \Box_r -formulae indeed follow from the definition of φ -atom, the definition of the labelling function, and the definition of R_φ , respectively. However, to obtain a model for φ we must also guarantee the truth of \Diamond_r -formulae. To this end, we introduce the notion of fulfilling LIS.

Definition 3.7. A φ -labelled interval structure $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ is *fulfilling* if and only if, for every temporal formula $\Diamond_r \psi \in \text{TF}(\varphi)$ and every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, if $\Diamond_r \psi \in \mathcal{L}([d_i, d_j])$, then there exists $d_k \geq d_j$ 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 interval $[d_0, d_0]$ labelled by $\Diamond_r \varphi$. The implication from left to right is straightforward; the opposite implication is proved by induction on the structure of the formula.

Theorem 3.8. *A formula φ is satisfiable if and only if there exists a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ with $\Diamond_r \varphi \in \mathcal{L}([d_0, d_0])$.*

Proof. Let φ be a satisfiable formula and let $\mathbf{M} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ be a model for it. We define a LIS $\mathbf{L}_\mathbf{M} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L}_\mathbf{M} \rangle$ such that for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, $\mathcal{L}_\mathbf{M}([d_i, d_j]) = \{\psi \in \text{CL}(\varphi) : \mathbf{M}, [d_i, d_j] \Vdash \psi\}$. It is immediate that $\mathbf{L}_\mathbf{M}$ is a fulfilling LIS and $\Diamond_r \varphi \in \mathcal{L}_\mathbf{M}([d_0, d_0])$.

As for the opposite implication, let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS with $\Diamond_r \varphi \in \mathcal{L}([d_0, d_0])$. We define a model $\mathbf{M}_\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V}_\mathbf{L} \rangle$ such that for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$ and every propositional letter $p \in AP$, $[d_i, d_j] \in \mathcal{V}_\mathbf{L}(p)$ if and only if $p \in \mathcal{L}([d_i, d_j])$. We prove by induction on the structure of φ that for every $\psi \in \text{CL}(\varphi)$ and every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, $\mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \psi$ if and only if $\psi \in \mathcal{L}([d_i, d_j])$. Since $\Diamond_r \varphi \in \mathcal{L}([d_0, d_0])$, we can conclude that $\mathbf{M}_\mathbf{L}, [d_0, d_0] \Vdash \Diamond_r \varphi$.

- If ψ is the propositional letter p , then $p \in \mathcal{L}([d_i, d_j]) \stackrel{\mathcal{V}_\mathbf{L} \text{ def.}}{\iff} [d_i, d_j] \in \mathcal{V}_\mathbf{L}(p) \iff \mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash p$.
- If ψ is the formula $\neg \xi$, then $\neg \xi \in \mathcal{L}([d_i, d_j]) \stackrel{\text{atom def.}}{\iff} \xi \notin \mathcal{L}([d_i, d_j]) \stackrel{\text{ind. hyp.}}{\iff} \mathbf{M}_\mathbf{L}, [d_i, d_j] \not\Vdash \xi \iff \mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \neg \xi$.
- If ψ is the formula $\xi_1 \vee \xi_2$, then $\xi_1 \vee \xi_2 \in \mathcal{L}([d_i, d_j]) \stackrel{\text{atom def.}}{\iff} \xi_1 \in \mathcal{L}([d_i, d_j])$ or $\xi_2 \in \mathcal{L}([d_i, d_j]) \stackrel{\text{ind. hyp.}}{\iff} \mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \xi_1$ or $\mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \xi_2 \iff \mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \xi_1 \vee \xi_2$.
- If ψ is the modal constant π , then $\pi \in \mathcal{L}([d_i, d_j]) \stackrel{\mathcal{L} \text{ def.}}{\iff} d_i = d_j \iff \mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \pi$.
- Let ψ be the formula $\Diamond_r \xi$. Suppose that $\Diamond_r \xi \in \mathcal{L}([d_i, d_j])$. 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 $\mathbf{M}_\mathbf{L}, [d_j, d_k] \Vdash \xi$, and hence $\mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \Diamond_r \xi$. As for the opposite implication, assume by contradiction that $\mathbf{M}_\mathbf{L}, [d_i, d_j] \Vdash \Diamond_r \xi$, but $\Diamond_r \xi \notin \mathcal{L}([d_i, d_j])$. By atom definition, this implies that $\neg \Diamond_r \xi = \Box_r \neg \xi \in \mathcal{L}([d_i, d_j])$. By definition of LIS, we have that, for every $d_k \geq d_j$, $\mathcal{L}([d_i, d_j]) R_\varphi$

$\mathcal{L}([d_j, d_k])$, and thus $\neg\xi \in \mathcal{L}([d_j, d_k])$. By inductive hypothesis, this implies that $\mathbf{M}_{\mathbf{L}}, [d_j, d_k] \Vdash \neg\xi$ for every $d_k \geq d_j$, and hence $\mathbf{M}_{\mathbf{L}}, [d_i, d_j] \Vdash \Box_r \neg\xi$, which contradicts the hypothesis that $\mathbf{M}_{\mathbf{L}}, [d_i, d_j] \Vdash \Diamond_r \xi$. \square

Theorem 3.8 reduces the satisfiability problem for φ to the problem of finding a fulfilling LIS with the initial interval $[d_0, d_0]$ labelled by $\Diamond_r \varphi$. From now on, we say that a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ satisfies φ if and only if $\Diamond_r \varphi \in \mathcal{L}([d_0, d_0])$.

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 labelling 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 $\Diamond_r \psi \in \text{TF}(\varphi)$, $\Diamond_r \psi \in \mathcal{L}([d_i, d_j])$ iff $\Diamond_r \psi \in \mathcal{L}([d_k, d_j])$ (it easily follows from Definitions 3.4, 3.5, and 3.6). The same holds for universal formulae $\Box_r \psi$.

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

Definition 3.9. Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ and $d \in D$, we denote by $\text{REQ}^{\mathbf{L}}(d)$ the set of all and only the temporal formulae belonging to the labelling 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 $|\text{REQ}_{\varphi}|$ is equal to $2^{\frac{\text{TF}(\varphi)}{2}}$.

Definition 3.10. Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$, a set of points $D' \subseteq D$, and a set of temporal formulae $\mathcal{R} \subseteq \text{TF}(\varphi)$, we say that \mathcal{R} occurs n times in D' if and only if there exist exactly n distinct points $d_{i_1}, \dots, d_{i_n} \in D'$ such that $\text{REQ}^{\mathbf{L}}(d_{i_j}) = \mathcal{R}$, for all $1 \leq j \leq n$.

We describe the process of removing a point from a LIS. Given $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ and $d \in D$, let \mathbf{L}_{-d} be the set of all LIS $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L}' \rangle$ such that $D' = D \setminus \{d\}$ and $\text{REQ}^{\mathbf{L}'}(\bar{d}) = \text{REQ}^{\mathbf{L}}(\bar{d})$, for all $\bar{d} \in D \setminus \{d\}$. \mathbf{L} and \mathbf{L}' do not necessarily agree on the labeling of intervals, but they agree on the sets of requests of points.

Given a fulfilling LIS \mathbf{L} and a point d , it is not guaranteed that \mathbf{L}_{-d} contains a fulfilling LIS. The removal of d indeed causes the removal of all intervals either

beginning or ending at it and thus there can be a point $\bar{d} < d$ such that there exists a formula $\diamond_r \psi \in \text{REQ}^{\mathbf{L}}(\bar{d})$ which is fulfilled in \mathbf{L} , but not in any $\mathbf{L}' \in \mathbf{L}_{-d}$. The following lemma provides a sufficient condition for preserving the fulfilling property when removing a point from \mathbf{L} .

Lemma 3.11. *Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS satisfying φ and let $m = \frac{|\text{TF}(\varphi)|}{2}$. If there exists a point $d_e > d_0$ such that there exist at least m distinct points $d > d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$, then there is one fulfilling LIS $\bar{\mathbf{L}} \in \mathbf{L}_{-d_e}$ that satisfies φ .*

Proof. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS satisfying φ and let $d_e > d_0$ be a point such that there exist at least m distinct points $d > d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$.

A fulfilling LIS $\bar{\mathbf{L}} \in \mathbf{L}_{-d_e}$ can be obtained as follows. Let $\mathbb{D}' = \langle D \setminus \{d_e\}, < \rangle$ and $\mathcal{L}' = \mathcal{L}|_{\mathbb{I}(\mathbb{D}')^+}$ (the restriction of \mathcal{L} to the intervals on \mathbb{D}'). $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L}' \rangle$ is obviously a LIS, but it is not necessarily a fulfilling one. The removal of d_e causes the removal of all intervals either beginning or ending at d_e . While the removal of intervals beginning at d_e is not critical (intervals ending at d_e are removed as well), there can be some points $d < d_e$ such that some formulae $\diamond_r \psi \in \text{REQ}^{\mathbf{L}'}(d)$ are fulfilled in \mathbf{L} , but they are not fulfilled in \mathbf{L}' anymore. We fix such defects (if any) one-by-one by properly redefining \mathcal{L}' . Let $d < d_e$ and $\diamond_r \psi \in \text{REQ}^{\mathbf{L}'}(d)$ be such that $\psi \in \mathcal{L}([d, d_e])$ and there exists no $d' \in D \setminus \{d_e\}$ such that $\psi \in \mathcal{L}'([d, d'])$. Since $\text{REQ}^{\mathbf{L}'}(d)$ contains at most m \diamond_r -formulae, there exists at least one point $d_j > d_e$ such that the atom $\mathcal{L}'([d, d_j])$ either fulfills no \diamond_r -formulae or it fulfills only \diamond_r -formulae which are also fulfilled by some other atom $\mathcal{L}'([d, d'])$. Let d_j one of such “useless” points. We can redefine $\mathcal{L}'([d, d_j])$ by putting $\mathcal{L}'([d, d_j]) = \mathcal{L}([d, d_e])$, thus fixing the problem with $\diamond_r \psi \in \text{REQ}^{\mathbf{L}'}(d)$. Notice that, since $\text{REQ}^{\mathbf{L}'}(d_j) = \text{REQ}^{\mathbf{L}}(d_e)$, such a change has no impact on the right neighboring intervals of $[d, d_j]$. In a similar way, we can fix the other possible defects caused by the removal of d_e . Let $\bar{\mathbf{L}} = \langle \bar{\mathbb{D}}, \mathbb{I}(\bar{\mathbb{D}}), \bar{\mathcal{L}} \rangle$ be the resulting LIS. It is immediate that it is fulfilling and that it satisfies φ . \square

Lemma 3.11 can be exploited to provide a bound on the size of finite fulfilling LISs, as shown by the following theorem.

Theorem 3.12. *Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a finite fulfilling LIS that satisfies φ and let $m = \frac{|\text{TF}(\varphi)|}{2}$. Then there exists a finite fulfilling LIS $\bar{\mathbf{L}} = \langle \bar{\mathbb{D}}, \mathbb{I}(\bar{\mathbb{D}})^+, \bar{\mathcal{L}} \rangle$ that satisfies φ such that, for every $\bar{d}_i \in \bar{D}$, $\text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_i)$ occurs at most m times in $\bar{D} \setminus \{d_0\}$.*

Proof. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a finite fulfilling LIS that satisfies φ . If for every $d_i \in D$, $\text{REQ}^{\mathbf{L}}(d_i)$ occurs at most m times in $D \setminus \{d_0\}$, 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 $\mathcal{R}_0 = \{\text{REQ}_1, \text{REQ}_2, \dots, \text{REQ}_k\}$ be the (arbitrarily ordered) finite set of all and only the sets of requests that occur more than m times in $D \setminus \{d_0\}$. We show how to turn \mathbf{L}_0 into a fulfilling LIS $\mathbf{L}_1 = \langle \mathbb{D}_1, \mathbb{I}(\mathbb{D}_1)^+, \mathcal{L}_1 \rangle$ satisfying φ , which,

unlike \mathbf{L}_0 , contains exactly m points $d \in D_1 \setminus \{d_0\}$ such that $\text{REQ}^{\mathbf{L}_1}(d) = \text{REQ}_1$. Such a fulfilling LIS can be obtained as follows. Let d_e be the smallest point in $\mathbb{D} \setminus \{d_0\}$ such that $\text{REQ}^{\mathbf{L}}(d_e) = \text{REQ}_1$. Since REQ_1 occurs more than m times in $\mathbb{D} \setminus \{d_0\}$ we have that there exists $n \geq m$ points $d > d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e) = \text{REQ}_1$. Hence, by Lemma 3.11, there exists a fulfilling LIS $\mathbf{L}' \in \mathbf{L}_{-d_e}$ satisfying φ . We repeat the application of Lemma 3.11 until we obtain a fulfilling LIS \mathbf{L}_1 that satisfies φ and such that REQ_1 occurs exactly m times in $\mathbb{D} \setminus \{d_0\}$.

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 . \square

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

Definition 3.13. An infinite LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ is *ultimately periodic*, with prefix l and period $p > 0$, if and only if for all $i > l$, $\text{REQ}^{\mathbf{L}}(d_i) = \text{REQ}^{\mathbf{L}}(d_{i+p})$.

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

1. for every pair of points $\bar{d}_i, \bar{d}_j \in \bar{D}$, with $\bar{d}_0 \leq \bar{d}_i \leq \bar{d}_l$ and $\bar{d}_j > \bar{d}_l$, $\text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_i) \neq \text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_j)$, that is, points belonging to the prefix and points belonging to the period have different sets of requests;
2. for every $\bar{d}_i \in \bar{D}$, with $\bar{d}_0 \leq \bar{d}_i \leq \bar{d}_l$, $\text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_i)$ occurs at most m times in $\{\bar{d}_1, \dots, \bar{d}_l\}$;
3. for every pair of points $\bar{d}_i, \bar{d}_j \in \bar{D}$, with $\bar{d}_{l+1} \leq \bar{d}_i, \bar{d}_j \leq \bar{d}_{l+p}$, if $i \neq j$, then $\text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_i) \neq \text{REQ}^{\bar{\mathbf{L}}}(\bar{d}_j)$.

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

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

We build an infinite ultimately periodic LIS $\bar{\mathbf{L}}$, with prefix $l \leq m \cdot |\text{Fin}(\mathbf{L})| + 1$ and period $p = |\text{Inf}(\mathbf{L})|$, that satisfies φ and respects Conditions 1 - 3, as follows.

1. Let d_l be the greatest point in D such that $\text{REQ}^{\mathbf{L}}(d_l) \in \text{Fin}(\mathbf{L})$. The set $\{d_0, \dots, d_l\}$ will be the prefix of $\bar{\mathbf{L}}$. By repeatedly applying Lemma 3.11 we can remove from the prefix all points d such that $\text{REQ}^{\mathbf{L}}(d) \in \text{Inf}(\mathbf{L})$.

2. Once we have removed all points d such that $\text{REQ}^{\mathbf{L}}(d) \in \text{Inf}(\mathbf{L})$, we can exploit Lemma 3.11 to remove from the prefix all points d' such that $\text{REQ}^{\mathbf{L}}(d')$ occurs more than m times in $\{d_1, \dots, d_i\}$.
3. To turn \mathbf{L} into an ultimately periodic LIS respecting Condition 3, we must show how to define the period of \mathbf{L} . We first collect and (arbitrarily) enumerate the elements of $\text{Inf}(\mathbf{L})$, that is, let $\text{Inf}(\mathbf{L}) = \{\text{REQ}_0, \dots, \text{REQ}_{p-1}\}$. The cardinality p of $\text{Inf}(\mathbf{L})$ will be the period of $\bar{\mathbf{L}}$. We inductively define $\bar{\mathbf{L}}$ in such a way that, for all $k \geq 0$, $\text{REQ}^{\bar{\mathbf{L}}}(d_{l+1+k}) = \text{REQ}_{k \bmod p}$. Let $k = 0$, and consider $\text{REQ}^{\mathbf{L}}(d_{l+1})$: if $\text{REQ}^{\mathbf{L}}(d_{l+1}) = \text{REQ}_0$, we are done. Otherwise, let $d_h > d_{l+1}$ be the first occurrence of REQ_0 after d_{l+1} . We have that, for every point $d_{l+1} \leq d' < d_h$, there exist sufficiently many points $d'' > d_h$ such that $\text{REQ}^{\mathbf{L}}(d'') = \text{REQ}^{\mathbf{L}}(d')$. Hence, by Lemma 3.11, there exists a LIS \mathbf{L}_0 where all points $d_{l+1} \leq d' < d_h$ have been removed. Thus, \mathbf{L}_0 is such that $\text{REQ}^{\mathbf{L}_0}(d_{l+1}) = \text{REQ}^{\bar{\mathbf{L}}}(d_h) = \text{REQ}_0$. Now, let $k > 0$ and suppose that $\mathbf{L}_{k-1} = \langle \mathbb{D}_{k-1}, \mathbb{I}(\mathbb{D}_{k-1})^+, \mathcal{L}_{k-1} \rangle$ respect the condition for all $h < k$. We can proceed as in the case of $k = 0$ and define a LIS $\mathbf{L}_k = \langle \mathbb{D}_k, \mathbb{I}(\mathbb{D}_k)^+, \mathcal{L}_k \rangle$ such that $\text{REQ}^{\mathbf{L}_k}(d_{l+1+k}) = \text{REQ}_{k \bmod p}$. \square

3.3 The complexity of the satisfiability problem for RPNL^{π+}

In this section we provide a precise characterization of the computational complexity of the satisfiability problem for RPNL^{π+}.

3.3.1 An upper bound to the computational complexity

A decision procedure for RPNL^{π+} can be derived from the results of Section 3.2 in a straightforward way. Theorems 3.12 and 3.14 indeed provide a bound on the size of the LISs to be checked:

- by Theorem 3.12, we have that if there exists a finite LIS satisfying φ , then there exists a finite one of size less than or equal to $|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1$ which satisfies φ ;
- by Theorem 3.14, we have that if there exists an infinite LIS satisfying φ , then there exists an ultimately periodic one, with prefix $l \leq |\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1$ and period $p \leq |\text{REQ}_\varphi|$, which satisfies φ .

A simple decision algorithm to check the satisfiability of an RPNL^{π+} formula φ that nondeterministically guesses a LIS \mathbf{L} satisfying it can be defined as follows.

First, the algorithm guesses the set $\text{Inf}(\mathbf{L}) = \{\text{REQ}_1, \dots, \text{REQ}_p\}$, with $\text{Inf}(\mathbf{L}) \subseteq \text{REQ}_\varphi$, of the sets of requests that occur infinitely often in \mathbf{L} . 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 $[d_0, d_0]$, taking an atom $A_{[d_0, d_0]}$ that includes $\diamond_r \varphi$ and π ($[d_0, d_0]$ is a point interval), and it initializes a counter c to 0. If $c < l$, then it guesses the labelling of the intervals ending in d_1 , that is, it associates an atom $A_{[d_0, d_1]}$ (that does not contain π) with $[d_0, d_1]$ and an atom $A_{[d_1, d_1]}$ (containing π) with $[d_1, d_1]$ such that $A_{[d_0, d_0]} R_\varphi A_{[d_0, d_1]} R_\varphi A_{[d_1, d_1]}$, with $\text{REQ}^L(d_1) \in \text{REQ}_\varphi \setminus \text{Inf}(\mathbf{L})$, 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 $A_{[d_i, d_j]}$, $A_{[d_k, d_i]} R_\varphi A_{[d_i, d_j]}$ and that $\pi \in A_{[d_i, d_j]}$ if and only if $d_i = d_j$. For each point d_j , it must guarantee that $\text{REQ}^L(d_j) \in \text{REQ}_\varphi \setminus \text{Inf}(\mathbf{L})$ and that $\text{REQ}^b L(d_j)$ occurs at most $\frac{|\text{TF}(\varphi)|}{2}$ times in $\{d_1, \dots, d_j\}$.

When c reaches the value l , two cases are possible. If $p = 0$, then d_l is the last point of the finite LIS \mathbf{L} , 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 $\diamond_r \psi \in A_{[d_i, d_j]}$, 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' and a set $\text{REQ}_h \in \text{Inf}(\mathbf{L})$ such that ψ and $\neg\pi$ belong to A' , $\text{REQ}_h = A' \cap \text{TF}(\varphi)$ and $A_{[d_i, d_j]} R_\varphi A'$;
- for every atom $A_{[d_i, d_j]}$ in the prefix and for every set $\text{REQ}_h \in \text{Inf}(\mathbf{L})$, it checks if there exists an atom A' such that $\pi \notin A'$, $\text{REQ}_h = A' \cap \text{TF}(\varphi)$, and $A_{[d_i, d_j]} R_\varphi A'$;
- for every set $\text{REQ}_h \in \text{Inf}(\mathbf{L})$ it checks if there exists an atom A_h such that $\pi \in A_h$, $\text{REQ}_h = A_h \cap \text{TF}(\varphi)$, and $\text{REQ}_h R_\varphi A_h$;
- for every set $\text{REQ}_h \in \text{Inf}(\mathbf{L})$ and for every formula $\diamond_r \psi \in \text{REQ}_h$ such that $\psi \notin A_h$, it checks if there exists an atom A' and a set $\text{REQ}_k \in \text{Inf}(\mathbf{L})$ such that ψ and $\neg\pi$ belong to A' , $\text{REQ}_k = A' \cap \text{TF}(\varphi)$, and $\text{REQ}_h R_\varphi A'$;
- for every pair of sets $\text{REQ}_h, \text{REQ}_k \in \text{Inf}(\mathbf{L})$, it checks if there exists an atom A' such that $\pi \notin A'$, $\text{REQ}_k = A' \cap \text{TF}(\varphi)$, and $\text{REQ}_h R_\varphi A'$.

By Theorems 3.12 and 3.14, 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 $d_0 \leq d_j \leq d_l$, the algorithm guesses exactly $j + 1$ atoms $A_{[d_i, d_j]}$;
3. checking for the fulfillment 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 $|\text{REQ}_\varphi|$ is exponential in it.

Hence, if $|\varphi| = n$, the number of guessed sets in $\text{Inf}(\mathbf{L})$ plus number of guessed atoms in the prefix is bounded by

$$\begin{aligned} |\text{REQ}_\varphi| + \sum_{i=0}^{|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2} + 1} i + 1 &= O\left(|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2}\right)^2 \\ &= \left(2^{O(n)} \cdot O(n)\right)^2 = 2^{2 \cdot O(n)} \cdot O(n^2) \\ &= 2^{O(n)} \cdot O(n^2) = 2^{O(n)}, \end{aligned}$$

that is, it is exponential in the length of φ . This implies that the satisfiability problem for $\text{RPNL}^{\pi+}$ can be solved by the above nondeterministic algorithm in nondeterministic exponential time.

Theorem 3.15. *The satisfiability problem for $\text{RPNL}^{\pi+}$, over natural numbers or over finite linear orderings, is in NEXPTIME.*

3.3.2 A lower bound to the computational complexity

We now provide a NEXPTIME lower bound for the complexity of the satisfiability problem for $\text{RPNL}^{\pi+}$ by reducing to it the exponential tiling problem, which is known to be NEXPTIME-complete [BGG97].

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 $\tau : N(m) \rightarrow C$ 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 $\text{RPNL}^{\pi+}$ 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 the condition that every point of the grid is tiled by exactly one color and that the colors respect the adjacency conditions. Intervals are exploited to express relations between pairs of points.

Theorem 3.16. *The satisfiability problem for $\text{RPNL}^{\pi+}$, over natural numbers, is NEXPTIME-hard.*

Proof. Given a domino system $\mathcal{D} = \langle C, H, V \rangle$, we build an $\text{RPNL}^{\pi+}$ 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 $\tau : N(2^n) \rightarrow C$ in the following way. First, we associate with every point $z = (x, y) \in N(2^n)$ a $2n$ -bit word $(z_{2n-1}z_{2n-2}\dots z_1z_0) \in \{0, 1\}^{2n}$ 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:

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

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:

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

To ease the writing of the formula 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:

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

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 $0 \leq i \leq 2n - 1$, and ZH_i^* , with $n \leq i \leq 2n - 1$, as follows:

$$\begin{aligned} & \square_r \square_r \left(Z_0^* \wedge \bigwedge_{i=1}^{2n-1} (Z_i^* \leftrightarrow (Z_{i-1}^* \wedge Z_{i-1})) \right) \\ & \square_r \square_r \left(ZH_n^* \wedge \bigwedge_{i=n+1}^{2n-1} (ZH_i^* \leftrightarrow (ZH_{i-1}^* \wedge Z_{i-1})) \right). \end{aligned}$$

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):

$$\begin{aligned}
& \bigwedge_{i=0}^{2n-1} (\diamond_r Z_i \rightarrow \square_r Z_i) \quad \wedge \quad \bigwedge_{i=0}^{2n-1} \square_r (\diamond_r Z_i \rightarrow \square_r Z_i) \\
& \bigwedge_{i=0}^{2n-1} (\diamond_r Z_i^* \rightarrow \square_r Z_i^*) \quad \wedge \quad \bigwedge_{i=0}^{2n-1} \square_r (\diamond_r Z_i^* \rightarrow \square_r Z_i^*) \\
& \bigwedge_{i=n}^{2n-1} (\diamond_r ZH_i^* \rightarrow \square_r ZH_i^*) \quad \wedge \quad \bigwedge_{i=n}^{2n-1} \square_r (\diamond_r ZH_i^* \rightarrow \square_r ZH_i^*) \\
& \bigwedge_{c \in C} (\diamond_r Z_c \rightarrow \square_r Z_c) \quad \wedge \quad \bigwedge_{c \in C} \square_r (\diamond_r Z_c \rightarrow \square_r Z_c) \\
& \bigwedge_{i=0}^{2n-1} \square_r (T_i \leftrightarrow \square_r Z_i) \quad \wedge \quad \bigwedge_{i=0}^{2n-1} \square_r \square_r (T_i \leftrightarrow \square_r Z_i) \\
& \bigwedge_{c \in C} \square_r (T_c \leftrightarrow \square_r Z_c) \quad \wedge \quad \bigwedge_{c \in C} \square_r \square_r (T_c \leftrightarrow \square_r Z_c)
\end{aligned}$$

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 $0 \leq j \leq 2n - 1$ 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):

$$\square_r \left(\diamond_r \neg (Z_{2n-1}^* \wedge Z_{2n-1}) \rightarrow \diamond_r \bigwedge_{i=0}^{2n-1} (T_i \leftrightarrow (Z_i \oplus Z_i^*)) \right).$$

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 $(2^n - 1, 2^n - 1)$:

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

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 $N(2^n)$:

$$\begin{aligned}\psi_H &:= \bigwedge_{i=n}^{2n-1} (Z_i \leftrightarrow T_i) \wedge \bigwedge_{i=0}^{n-1} (T_i \leftrightarrow (Z_i \oplus Z_i^*)) \\ \psi_V &:= \bigwedge_{i=0}^{n-1} (Z_i \leftrightarrow T_i) \wedge \bigwedge_{i=n}^{2n-1} (T_i \leftrightarrow (Z_i \oplus ZH_i^*))\end{aligned}$$

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

$$\square_r \square_r \left(\left(\psi_H \rightarrow \bigvee_{(c,c') \in H} Z_c \wedge T_{c'} \right) \wedge \left(\psi_V \rightarrow \bigvee_{(c,c') \in V} Z_c \wedge T_{c'} \right) \right).$$

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

$$\square_r \square_r \left(\bigvee_{c \in C} Z_c \wedge \bigvee_{c \in C} T_c \right),$$

where \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 $\alpha \wedge \beta \wedge \gamma \wedge \delta \wedge \varepsilon \wedge \zeta$. 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 $z, t \in N(2^n)$, $[z, t] \in \mathcal{V}(Z_i)$ if and only if $z_i = 1$ and $[z, t] \in \mathcal{V}(T_i)$ if and only if $t_i = 1$. Moreover, $[z, t] \in \mathcal{V}(Z_c)$ (resp., $[z, t] \in \mathcal{V}(T_c)$) 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 $\mathbf{M} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ is a model of φ , that is, $\mathbf{M}, [d_0, d_1] \Vdash \varphi$.

Conversely, let $\mathbf{M} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ 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 $f : N(2^n) \rightarrow D$ 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 $Z_{2n-1}, Z_{2n-2}, \dots, Z_0$ 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 $d' \geq f(x, y)$. It is not difficult to prove that τ defines a tiling of $N(2^n)$. \square

From Theorems 3.15 and 3.16, we have the following corollary.

Corollary 3.17. *The satisfiability problem for RPNL^{π+}, over natural numbers, is NEXPTIME-complete.*

3.4 A tableau-based decision procedure for RPNL^{π+}

In this section, we define a tableau-based decision procedure for RPNL^{π+}, 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.

3.4.1 The tableau method

We first define the structure of a tableau for an RPNL^{π+} formula and then we show how to construct it. A tableau for RPNL^{π+} 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 $d_i \leq d_j$, 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 $\text{REQ}^B : D_B \rightarrow 2^{\text{TF}(\varphi)}$ as follows. For every $d_i \in D_B$, $\text{REQ}^B(d_i) = (\bigcap_j A_j) \cap \text{TF}(\varphi)$, where n_j is a node such that $\nu(n_j) = \langle [d_j, d_i], A_j \rangle$ and $d_0 \leq d_j \leq d_i$. Moreover, given a node $n \in B$, with decoration $\langle [d_i, d_j], A \rangle$, and an existential formula $\diamond_r \psi \in A$, we say that $\diamond_r \psi$ 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 $\pi \notin A'$, 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 $d_j \leq d_k$, 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 $\pi \in A''$ iff $d_i = d_j$, $A R_\varphi A''$ and $\text{REQ}^B(d_k) = A'' \cap \text{TF}(\varphi)$, and expand B to $B \cdot n''$, with $\nu(n'') = \langle [d_j, d_k], A'' \rangle$.

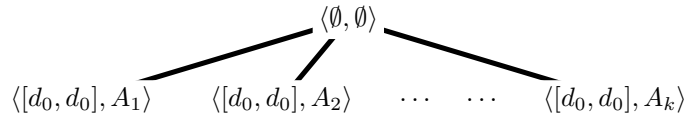
Both rules add a new node to the branch B . However, while the step rule decorates such a node with a new interval ending at a new point d_{k+1} , the fill-in rule decorates it with a new interval whose endpoints were already in D_B .

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 $D_B = \{d_0, d_1, \dots, d_k\}$, we say that B is *blocked* if $\text{REQ}^B(d_k)$ 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 $d_j \leq d_k$, 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, \dots, A_k be all and only the atoms containing $\diamond_r \varphi$ and π , and such that $A_i R_\varphi A_i$ for all $1 \leq i \leq k$. 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 $\diamond_r \psi \in A(n)$, there exists a node $n' \in B$ fulfilling $\diamond_r \psi$ (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 $\text{REQ}^B(d_i) = \text{REQ}^B(d_k)$, and the following conditions hold:
 - (a) for every node $n \in B$ and every formula $\diamond_r \psi \in A(n)$ not fulfilled on B , there exist a point $d_i \leq d_l \leq d_k$ and an atom A' such that $\pi \notin A'$, $\psi \in A'$, $A(n) R_\varphi A'$, and $\text{REQ}^B(d_l) = A' \cap \text{TF}(\varphi)$;
 - (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 $\pi \notin A'$, $A(n) R_\varphi A'$ and $\text{REQ}^B(d_m) = A' \cap \text{TF}(\varphi)$.

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} .

We conclude the section by showing how the proposed method works on the simple case of the formula $\varphi = \Box_r \pi$. The set of φ -atoms is the following one:

$$\begin{array}{ll}
A_0 & = \{ \diamond_r \Box_r \pi, \Box_r \pi, \pi \} & A_4 & = \{ \Box_r \diamond_r \neg \pi, \Box_r \pi, \pi \} \\
A_1 & = \{ \diamond_r \Box_r \pi, \Box_r \pi, \neg \pi \} & A_5 & = \{ \Box_r \diamond_r \neg \pi, \Box_r \pi, \neg \pi \} \\
A_2 & = \{ \diamond_r \Box_r \pi, \diamond_r \neg \pi, \pi \} & A_6 & = \{ \Box_r \diamond_r \neg \pi, \diamond_r \neg \pi, \pi \} \\
A_3 & = \{ \diamond_r \Box_r \pi, \diamond_r \neg \pi, \neg \pi \} & A_7 & = \{ \Box_r \diamond_r \neg \pi, \diamond_r \neg \pi, \neg \pi \}
\end{array}$$

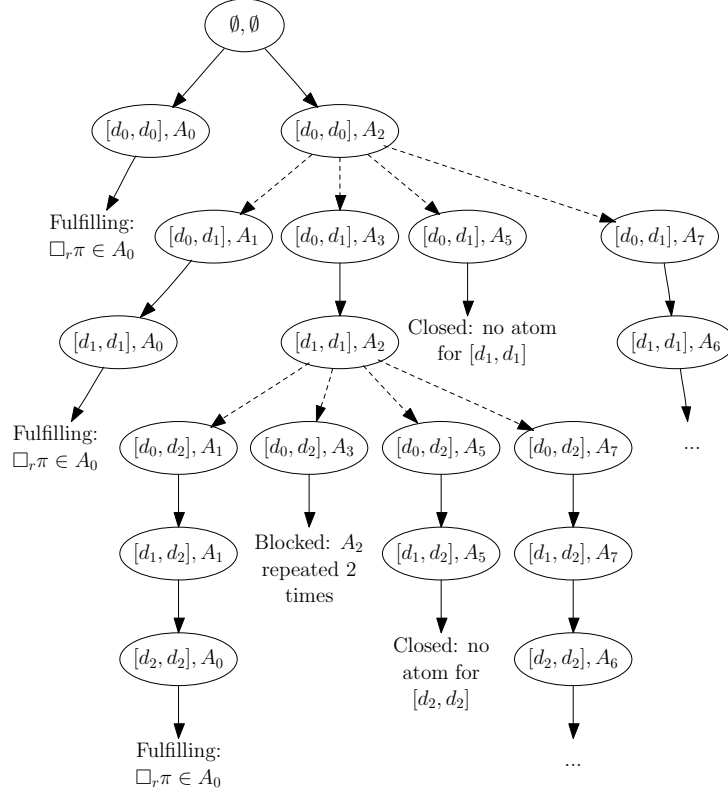
Figure 3.2, where dashed arrows represent applications of the step rule, depicts a portion of a tableau for φ which is sufficiently large to include a fulfilling branch, and thus to prove that φ is satisfiable. Indeed, it is easy to see that, over natural numbers, φ is satisfiable and it admits only *finite models*.

3.4.2 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 $|\text{REQ}_\varphi| \cdot \frac{|\text{TF}(\varphi)|}{2}$ 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);

Figure 3.2: A tableau for $\Box_r \pi$.

3. $|\text{TF}(\varphi)|$ is linear in the length of φ , while $|\text{REQ}_\varphi|$ is exponential in it.

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

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

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

Theorem 3.18 (Termination). *The tableau method for $\text{RPNL}^{\pi+}$ 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. \square

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 $\langle [d_0, d_0], A \rangle$ 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^{π+}, this allows us to conclude that the proposed tableau-based decision procedure is optimal.

3.4.3 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 3.8, 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 3.12 and 3.14.

Theorem 3.19 (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*). Let $\mathbf{L} = \langle \mathbb{D}_B, \mathbb{I}(\mathbb{D}_B)^+, \mathcal{L} \rangle$ be a LIS such that, for every $[d_i, d_j] \in \mathbb{I}(\mathbb{D}_B)^+$, $\mathcal{L}([d_i, d_j]) = A(n)$, where n is the unique node in B such that $\nu(n) = \langle [d_i, d_j], A(n) \rangle$. Since B is not closed, such a node n exists; its uniqueness follows from tableau rules. From the fact that B is a fulfilling branch it follows that for every node $n \in B$ and every existential formula $\diamond_r \psi \in A(n)$, there exists a node n' fulfilling $\diamond_r \psi$. 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 $\text{REQ}^B(d_i) = \text{REQ}^B(d_k)$. We build an ultimately periodic LIS $\mathbf{L} = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L} \rangle$ with prefix $l = i - 1$ and period $p = k - i$, as follows:

1. let $D' = \{d'_0(=d_0), d'_1(=d_1), \dots, d'_k(=d_k), d'_{k+1}, \dots\}$ be any set isomorphic to \mathbb{N} ;
2. for every $d'_1 \leq d'_j \leq d'_k$, put $\text{REQ}^{\mathbf{L}}(d'_j) = \text{REQ}^B(d_j)$;
3. for every $d'_j > d'_k$, put $\text{REQ}^{\mathbf{L}}(d'_j) = \text{REQ}^B(d_{i+(j-l) \bmod p})$;
4. for every pair of points d'_j, d'_m such that $d_0 \leq d'_j \leq d'_m \leq d'_k$, take the node n in B such that $\nu(n) = \langle [d_j, d_m], A \rangle$ and put $\mathcal{L}([d'_j, d'_m]) = A$;
5. for every point $d'_j > d'_k$, let $h = i + (j - l) \bmod p$. Take the node n in B such that $\nu(n) = \langle [d_h, d_h], A \rangle$ and put $\mathcal{L}([d'_j, d'_j]) = A$;
6. for every point $d'_j \in D'$ and every $\diamond_r \psi \in \text{REQ}^{\mathbf{L}}(d'_j)$ 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 $\text{REQ}^{\mathbf{L}}(d') = \text{REQ}^{\mathbf{L}}(d'_j)$. 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 $\text{REQ}^B(d_m) = A' \cap \text{TF}(\varphi)$. By the definition of \mathbf{L} , we have that there exist infinitely many points $d'_n \geq d'_k$ in D' such that $\text{REQ}^{\mathbf{L}}(d'_n) = \text{REQ}^B(d_m)$. 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 \diamond_r -formulae in $\text{REQ}^{\mathbf{L}}(d')$, 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;

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

Theorem 3.20 (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 $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS satisfying φ , whose existence is guaranteed by Theorem 3.8. Without loss of generality, we may assume that \mathbf{L} respects the constraints of Theorem 3.12, if it is finite, and of Theorem 3.14, 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

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

with $A = \mathcal{L}([d_0, d_0])$.

Let $h(\mathcal{T}) = i + 1$. By inductive hypothesis, there exists a branch B_i of length i that satisfies the property. Let $D_{B_i} = \{d_0, d_1, \dots, d_k\}$. 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 $\langle [d_j, d_i], A \rangle$, which the step rule has been applied to. By inductive hypothesis, $A = \mathcal{L}([d_j, d_i])$. Since \mathbf{L} is a LIS, $\mathcal{L}([d_j, d_i]) R_\varphi \mathcal{L}([d_l, d_{k+1}])$. Hence, there must exist 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 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 $\langle [d_j, d_i], A \rangle$ 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, $A = \mathcal{L}([d_j, d_i])$. Since \mathbf{L} is a LIS, $\mathcal{L}([d_j, d_i]) R_\varphi \mathcal{L}([d_l, d_k])$. Hence, there must exist 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 $\diamond_r \psi \in A(n)$, there exists a node n' fulfilling $\diamond_r \psi$. Hence, B is a fulfilling branch.
- *B is blocked.* This implies that $\text{REQ}^B(d_k)$ 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 3.12, we can assume \mathbf{L} to be infinite. Let d_j be the smallest point in D_B such that $\text{REQ}^B(d_j) = \text{REQ}^B(d_k)$. We have that \mathbf{L} is ultimately periodic, with prefix $l = j - 1$, since (by Theorem 3.14) 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 $\diamond_r \psi \in A$, 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 $\diamond_r \psi$ is fulfilled in B . Otherwise, there exists some point d_m , with $d_i \leq d_m \leq d_k$, such that $\text{REQ}^{\mathbf{L}}(d'') = \text{REQ}^{\mathbf{L}}(d_m)$. 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 $\text{REQ}^{\mathbf{L}}(d_m) \in \text{Inf}(\mathbf{L})$. Hence, there exist infinitely many points

d_n in \mathbf{L} such that $\text{REQ}^{\mathbf{L}}(d_m) = \text{REQ}^{\mathbf{L}}(d_n)$ 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. \square

3.5 A tableau-based decision procedure for RPNL^+ and RPNL^-

The tableau-based decision procedure for $\text{RPNL}^{\pi+}$ presented in the previous sections can be easily adapted to a decision procedure for RPNL^+ by simply ignoring the conditions on the π operator (that is not present in RPNL^+). In this section we briefly show how to adapt it to the case of RPNL^- . First of all, we need to define the notion of *strict* φ -labelled interval structure as follows.

Definition 3.21. A *strict* φ -labelled interval structure (strict LIS, for short) is a pair $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-, \mathcal{L} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle$ is a strict interval structure and $\mathcal{L} : \mathbb{I}(\mathbb{D})^- \rightarrow A_\varphi$ 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 3.8, 3.12, and 3.14 hold also for strict LISs. Furthermore, we can easily tailor the tableau-based decision method for $\text{RPNL}^{\pi+}$ to RPNL^- by ignoring the constraints on the π operator and by rewriting the fill-in rule as follows. We recall that, given a branch B , d_k is the last point of the ordering \mathbb{D}_B .

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 $\text{REQ}^B(d_k) = A'' \cap \text{TF}(\varphi)$, and expand B to $B \cdot n''$, with $\nu(n'') = \langle [d_j, d_k], A'' \rangle$.

The definition of initial tableau has to be modified as follows. Let φ be the formula to be checked for satisfiability and let A_1, \dots, A_k be all and only the atoms containing φ . The *initial tableau* for φ is composed by the empty root $\langle \emptyset, \emptyset \rangle$ with k immediate successors n_1, \dots, n_k such that, for each $1 \leq i \leq k$, n_i is labelled with $\langle [d_0, d_1], A_i \rangle$.

By contrast, the expansion strategy, the blocking condition 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 $\text{RPNL}^{\pi+}$.

Finally, to prove the optimality of the tableau for RPNL^- , we can exploit the reduction given in Section 3.3, provided that we replace \diamond_r by $\langle A \rangle$ and \square_r by $[A]$.

Theorem 3.22. *The satisfiability problem for RPNL^- , over natural numbers, is NEXPTIME-complete.*

4

The tableau method for BTNL

When defining a temporal logic, there are basically two possible choices for the underlying temporal structure. Either time is *linear* (at any time there is only one possible future) or it has a *branching*, tree-like structure (any time may have many different futures). In the case of point-based temporal logics, both these alternatives have been successfully explored, and several meaningful logics have been developed (we only mention the linear temporal logic LTL and its many variants, and the branching-time temporal logics CTL and CTL* [Eme90]). By contrast, interval-based temporal logics are usually interpreted over linear temporal structures. Even those interval logics which are interpreted over branching-time temporal structures, such as Halpern and Shoham's HS (in its original formulation) and Goranko, Montanari, and Sciavicco's branching CDT (BCDT⁺), only feature temporal operators that express properties of single timelines, with the only exception of Paech's Branching Regular Logic (BRL) [Pae89]. BRL is a branching-time interval logic with the locality assumption, whose operators quantify over different timelines. In [Pae89] the author provides a Gentzen-style system for BRL and she states some expressiveness and complexity results.

In this chapter we consider the branching-time interval neighborhood logic BTNL. We recall from Chapter 1 that such logic interleaves operators that quantify over possible timelines with operators that quantify over intervals belonging to a given timeline. Formulae of BTNL are interpreted over infinite trees where every path in the tree is isomorphic to $\langle \mathbb{N}, < \rangle$. Unlike the case of BRL, we do not impose any semantic restriction, such as locality, to get decidability. Putting together the tableau method for CTL [Eme90] and the one we developed for RPNL in Chapter 3, we have been able to devise a doubly-exponential tableau-based decision procedure for BTNL. This chapter is an extension and a revision of [BM05a].

4.1 Basic Notions

To check the satisfiability of a formula φ , we build a *tableau* for φ , whose nodes represent points of the infinite tree \mathbb{T} and whose edges represent the relation S connecting a point to its successors in the tree. We shall take advantage of such a construction to reduce the problem of finding a model for φ to the problem of testing whether the tableau satisfies some suitable properties or not. In contrast with the tableau method

we presented in Chapter 3 for RPNL and with the one we will discuss in Chapter 5 for PNL, in the tableau method for BTNL every node represent a *set* of intervals (corresponding to a layer of the structure in Figure 3.1) instead of a single interval.

Let $\mathbf{M}^+ = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+, \mathcal{V} \rangle$ be a model for φ and let t_j be a point in \mathbb{T} . We have that, given an interval $[t_i, t_j]$ ending in t_j , every right neighbor of it is also a right neighbor of every other interval $[t_k, t_j]$ ending in t_j . Hence, every *temporal formula* φ holds over $[t_i, t_j]$ if and only if it holds over every other interval $[t_k, t_j]$ ending in t_j . We denote by $\text{REQ}(t_j)$ the set of temporal formulae which hold over all intervals ending in t_j .

The building blocks for the tableau construction are φ -atoms. However, φ -atoms for BTNL are defined in a different way than atoms for the RPNL tableau. For every interval $[t_i, t_j] \in \mathbb{I}(\mathbb{T})^+$, we introduce a pair of sets of formulae $(R_{[t_i, t_j]}, C_{[t_i, t_j]})$, that we call an *atom* for φ (φ -atom for short). The set $R_{[t_i, t_j]}$ is a subset of $\text{REQ}(t_j)$, which collects the set of requests in $\text{REQ}(t_j)$ relevant to the interval $[t_i, t_j]$. In general, $R_{[t_i, t_j]}$ may differ from $R_{[t_i, t_k]}$ for $j \neq k$. The set $C_{[t_i, t_j]}$ contains all and only the formulae that (should) hold over $[t_i, t_j]$. We can associate with every point $t_j \in D$ the set of φ -atoms $\{(R_{[t_i, t_j]}, C_{[t_i, t_j]}) : t_i \leq t_j\}$, which includes all φ -atoms paired with intervals ending in t_j . These sets of atoms are the (macro)nodes of our tableau method for BTNL.

Let φ be a BTNL-formula to be checked for satisfiability and let AP be the set of its propositional variables. We define the *closure* $\text{CL}(\varphi)$ of φ as the set of all subformulae of $E \diamond_r \varphi$ and their negations (we identify $\neg \neg \psi$ with ψ), and the set of *temporal requests* of φ as the set $\text{TF}(\varphi)$ of all temporal formulae in $\text{CL}(\varphi)$. By induction on the structure of φ , it can be easily proved that $|\text{CL}(\varphi)| \leq 2 \cdot (|\varphi| + 1)$ and $|\text{TF}(\varphi)| \leq 2 \cdot (|\varphi| + 1)$.

We are now ready to formally define the key notion of φ -atom.

Definition 4.1. Let φ be a BTNL-formula. A φ -atom is a pair (R, C) , with $R \subseteq \text{TF}(\varphi)$ and $C \subseteq \text{CL}(\varphi)$, such that:

- for every $\psi \in \text{CL}(\varphi)$, $\psi \in C$ iff $\neg \psi \notin C$;
- for every $\psi_1 \vee \psi_2 \in \text{CL}(\varphi)$, $\psi_1 \vee \psi_2 \in C$ iff $\psi_1 \in C$ or $\psi_2 \in C$;
- for every $\psi \in \text{TF}(\varphi)$, if $\psi \in R$ then $\neg \psi \notin R$;
- for every $A \square_r \psi \in R$, $\psi \in C$;
- for every $E \square_r \psi \in R$, $\psi \in C$.

Formulae in C are called *current formulae*, while (temporal) formulae in R are called *active requests*. A φ -atom (R, C) is called a *point φ -atom* if and only if $\pi \in \text{CL}(\varphi)$ implies $\pi \in C$ and, for every $\psi \in \text{TF}(\varphi)$, $\psi \in R$ if and only if $\psi \in C$. Point atoms are used to represent point intervals.

As for the set R , it is worth pointing out that there may exist a φ -atom (R, C) and a temporal formula ψ such that neither ψ nor $\neg \psi$ belongs to R . We denote the set of all φ -atoms by \mathcal{A}_φ . It is not difficult to show that $|\mathcal{A}_\varphi| \leq 2^{2(|\varphi|+1)}$.

Atoms come into play in the proposed tableau method as follows. The method associates an atom (R, C) with any interval $[t_i, t_j]$. The set R includes all formulae of

the form $A\Box_r\psi$ that belong to $\text{REQ}(t_i)$ as well as some formulae of the forms $E\Box_r\psi$, $A\Diamond_r\psi$, and $E\Diamond_r\psi$ in $\text{REQ}(t_i)$; the set C includes all formulae $\psi \in \text{CL}(\varphi)$ which (should) hold over $[t_i, t_j]$. Moreover, for all formulae of the forms $A\Box_r\psi$ and $E\Box_r\psi$ in R , we put ψ into C , while for any formula of the forms $A\Diamond_r\psi$ and $E\Diamond_r\psi$ in R , it may happen that $\psi \in C$, but this is not necessarily the case.

Atoms are connected by the following binary relation.

Definition 4.2. Let X_φ be a binary relation over \mathcal{A}_φ such that, for every pair of atoms $(R, C), (R', C') \in \mathcal{A}_\varphi$, $(R, C) X_\varphi (R', C')$ if (and only if):

- $\pi \notin C'$;
- $R' \subseteq R$;
- for every $A\Box_r\psi \in R$, $A\Box_r\psi \in R'$;
- for every $A\Diamond_r\psi \in R$, $A\Diamond_r\psi \in R'$ iff $\neg\psi \in C$.

In the next section we shall show that for any pair of points $t_i \leq t_j$, the relation X_φ connects the atom associated with the interval $[t_i, t_j]$ to the atom associated with the interval $[t_i, t_{j+1}]$, where t_{j+1} is an R -successor of t_j . In particular, it will turn out that, if (R, C) is associated with the interval $[t_i, t_j]$, then for every formula $A\Box_r\psi \in \text{REQ}(t_i)$ and every atom (R', C') such that $(R, C) X_\varphi (R', C')$, we have that $A\Box_r\psi \in R'$ (and thus $\psi \in C'$), while for every formula $A\Diamond_r\psi \in R$, if $[t_i, t_j]$ satisfies ψ then $A\Diamond_r\psi \notin R'$. This guarantees that temporal requests of the form $A\Box_r\psi$ are propagated through X_φ -successors, while temporal requests of the form $A\Diamond_r\psi$ are discarded once that ψ has been satisfied by the set of current formulae of some atom.

4.2 Tableau construction

Nodes of the tableau for BTNL can be viewed as (maximal) collections of intervals ending at the same point t_j of the temporal domain. They are characterized by two components: a designated point atom representing the point interval $[t_j, t_j]$ and a set of atoms representing the proper intervals ending in t_j .

Definition 4.3. A *node* of the tableau is a pair $\mathbf{n} = \langle (R_{\mathbf{n}}, C_{\mathbf{n}}), M_{\mathbf{n}} \rangle$ where $(R_{\mathbf{n}}, C_{\mathbf{n}})$ is a point atom and $M_{\mathbf{n}}$ is a set of atoms such that, for any atom $(R, C) \in M_{\mathbf{n}}$:

- (i) $\pi \notin C$;
- (ii) for any $\psi \in \text{TF}(\varphi)$, $\psi \in C$ implies $\psi \in R_{\mathbf{n}}$.

We denote by \mathcal{N}_φ the set of all nodes that can be built from \mathcal{A}_φ and by $\text{Init}(\mathcal{N}_\varphi)$ the subset of all *initial nodes*, that is, the set $\{\langle (R, C), \emptyset \rangle \in \mathcal{N}_\varphi : E\Diamond_r\varphi \in C\}$. Furthermore, for any node \mathbf{n} , we denote by $\text{REQ}(\mathbf{n})$ the set $\{\psi \in \text{TF}(\varphi) : \psi \in R_{\mathbf{n}}\}$.

From Definition 4.3, it follows that $|\mathcal{N}_\varphi| \leq 2^{2^{-(|\varphi|+1)}}$. We can associate every node \mathbf{n} with a point $t_j \in T$. The atom $(R_{\mathbf{n}}, C_{\mathbf{n}})$ is thus associated with the point interval $[t_j, t_j]$, while every atom $(R, C) \in M_{\mathbf{n}}$ is associated with some interval $[t_i, t_j]$ such that $t_i < t_j$. Accordingly, we have that $\text{REQ}(\mathbf{n}) = \text{REQ}(t_j)$.

The relation between a node \mathbf{n} , associated with point t_j , and a node \mathbf{m} , associated with an S -successor t_{j+1} of t_j , as well as the relations between intervals ending in t_j and intervals ending in t_{j+1} (and, thus, between atoms in \mathbf{n} and atoms in \mathbf{m}), are graphically depicted in Figure 4.1. We have that for every interval $[t_i, t_j]$ (possibly, with $t_i = t_j$), there exists an interval $[t_i, t_{j+1}]$. Thus, \mathbf{m} should contain

- an atom (R'_n, C'_n) such that $(R_n, C_n) X_\varphi (R'_n, C'_n)$;
- for every atom $(R, C) \in M_n$, an atom (R', C') such that $(R, C) X_\varphi (R', C')$.

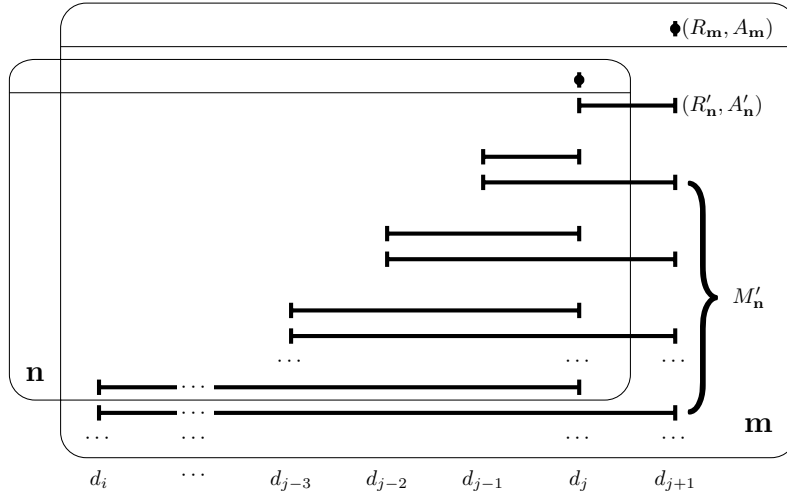


Figure 4.1: Connecting two nodes.

Definition 4.4. The *tableau* for a BTNL-formula φ is a (finite) directed graph $\mathcal{T}_\varphi = \langle \mathcal{N}_\varphi, \mathcal{S}_\varphi \rangle$, where for any pair of nodes $\mathbf{n} = \langle (R_n, C_n), M_n \rangle$ and $\mathbf{m} = \langle (R_m, C_m), M_m \rangle$, $(\mathbf{n}, \mathbf{m}) \in \mathcal{S}_\varphi$ if and only if $M_m = \{(R'_n, C'_n)\} \cup M'_n$, where

1. (R'_n, C'_n) is an atom such that $(R_n, C_n) X_\varphi (R'_n, C'_n)$;
2. for every $(R, C) \in M_n$, there exists $(R', C') \in M'_n$ such that $(R, C) X_\varphi (R', C')$;
3. for every $(R', C') \in M'_n$, there exists $(R, C) \in M_n$ such that $(R, C) X_\varphi (R', C')$.

Let $\mathbf{n}, \mathbf{m} \in \mathcal{N}_\varphi$. If $(\mathbf{n}, \mathbf{m}) \in \mathcal{S}_\varphi$, we say that \mathbf{m} is an \mathcal{S}_φ -*successor* of \mathbf{n} . We say that \mathbf{m} is an \mathcal{S}_φ -*descendant* of \mathbf{n} if there exists a (finite) path from \mathbf{n} to \mathbf{m} in \mathcal{T}_φ . Given a node $\mathbf{n} = \langle (R_n, C_n), M_n \rangle$ and an atom $(R, C) \in \mathcal{A}_\varphi$, we say that (R, C) *belongs to* \mathbf{n} (and we denote it with $(R, C) \in \mathbf{n}$) if $(R, C) = (R_n, C_n)$ or $(R, C) \in M_n$.

Definition 4.5. Given a (finite or infinite) path $\rho = \mathbf{n}_1 \mathbf{n}_2 \dots$ in \mathcal{T}_φ , an *atom path* in ρ is a sequence of atoms $(R_1, C_1)(R_2, C_2) \dots$ such that:

- for every $i \geq 1$, $(R_i, C_i) \in \mathbf{n}_i$;
- for every $i \geq 1$, $(R_i, C_i) X_\varphi (R_{i+1}, C_{i+1})$.

Given a node \mathbf{n} and an atom $(R, C) \in \mathbf{n}$, we say that the atom (R', C') is an X_φ -descendant of (R, C) if and only if there exists a node \mathbf{m} such that $(R', C') \in \mathbf{m}$, and there exists a path ρ from \mathbf{n} to \mathbf{m} such that there is an atom path from (R, C) to (R', C') in ρ .

Definition 4.6. An infinite path $\rho = \mathbf{n}_1 \mathbf{n}_2 \dots$ in \mathcal{T}_φ is a *fulfilling path* if and only if, for every $i \geq 1$, every atom $(R, C) \in \mathbf{n}_i$, and every formula $A \diamond_r \psi \in R$, either $\psi \in C$ or there exist \mathbf{n}_j , with $j > i$, and $(R', C') \in \mathbf{n}_j$ such that (R', C') is an X_φ -descendant of (R, C) in ρ and $\psi \in C'$.

Definition 4.7. A *substructure* is a subgraph $\langle \mathcal{N}, \mathcal{S} \rangle \subseteq \mathcal{T}_\varphi$ such that:

- there exists a node $\mathbf{n}_0 \in \mathcal{N} \cap \text{Init}(\mathcal{N}_\varphi)$ (*initial node*) such that all other nodes in \mathcal{N} are \mathcal{S} -reachable from it;
- for every node $\mathbf{n} \in \mathcal{N}$, there exists a fulfilling path (in $\langle \mathcal{N}, \mathcal{S} \rangle$) starting from \mathbf{n} .

Substructures represent *candidate models* for φ . The truth of formulae devoid of temporal operators and of formulae of the form $A \square_r \psi$, indeed, follows from Definition 4.1. Moreover, the truth of formulae of the form $A \diamond_r \psi$ follows from Definition 4.6. However, to obtain a model for φ we must also guarantee the truth of formulae of the forms $E \square_r \psi$ and $E \diamond_r \psi$. To this end, we introduce the notion of *fulfilling substructure*.

Definition 4.8. A substructure $\langle \mathcal{N}, \mathcal{S} \rangle \subseteq \mathcal{T}_\varphi$ is *fulfilling* if and only if, for every node $\mathbf{n} \in \mathcal{N}$ and every atom $(R, C) \in \mathbf{n}$, the following conditions hold:

- (F1) for every formula $E \diamond_r \psi \in R$, either $\psi \in C$ or there exist an \mathcal{S} -descendant \mathbf{m} of \mathbf{n} and an X_φ -descendant (R', C') of (R, C) in \mathbf{m} such that $\psi \in C'$;
- (F2) for every formula $E \square_r \psi \in R$, there exist a fulfilling path $\rho = \mathbf{n}_0 \mathbf{n}_1 \mathbf{n}_2 \dots$ and an atom path $(R_0, C_0)(R_1, C_1)(R_2, C_2) \dots$ in ρ such that: (i) $(R_0, C_0) = (R, C)$; (ii) $\mathbf{n}_0 = \mathbf{n}$; (iii) for every $i \geq 0$, $E \square_r \psi \in R_i$; (iv) for every formula $A \diamond_r \theta \in R_0$, there exists $j \geq 0$ such that $\theta \in C_j$.

Theorem 4.9. *If the formula φ is satisfiable (in an infinite tree), then there exists a fulfilling substructure $\langle \mathcal{N}, \mathcal{S} \rangle \subseteq \mathcal{T}_\varphi$.*

Proof. Let $\mathbf{M}^* = \langle \mathbb{T}^*, \mathbb{I}(\mathbb{T}^*)^+, \mathcal{V}^* \rangle$ be a model for φ and let $[t_0, t_1]$ be an interval such that $\mathbf{M}^*, [t_0, t_1] \Vdash \varphi$. Consider now the subtree \mathbb{T} of \mathbb{T}^* rooted at t_0 . Since BTNL features only future-time operators, the sub-model $\mathbf{M}^+ = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+, \mathcal{V} \rangle$ of \mathbf{M}^* generated by \mathbb{T} is a model of φ as well. Furthermore, we have that $\mathbf{M}^+, [t_0, t_0] \Vdash E \diamond_r \varphi$.

For every interval $[t_i, t_j] \in \mathbb{I}(\mathbb{T})^+$, we define an atom $(R_{[t_i, t_j]}, C_{[t_i, t_j]})$ as follows.

- $R_{[t_i, t_j]}$ contains exactly:
 - all formulae $A \square_r \psi \in \text{REQ}(t_i)$;
 - all formulae $A \diamond_r \psi \in \text{REQ}(t_i)$ such that for every $t_i \leq t_l < t_j$, $\mathbf{M}^+, [t_i, t_l] \Vdash \neg \psi$;

- all formulae $E\Box_r\psi \in \text{REQ}(t_i)$ such that there exists an infinite path $\rho = t_i t_{i+1} \dots t_j t_{j+1} \dots$, starting from t_i and containing t_j , such that $\mathbf{M}^+, [t_i, t_k] \Vdash \psi$ for every $t_k \in \rho$;
- all formulae $E\Diamond_r\psi \in \text{REQ}(t_i)$ such that there exists $t_k \geq t_j$ such that $\mathbf{M}^+, [t_i, t_k] \Vdash \psi$ and, for every $t_i \leq t_l < t_k$, $\mathbf{M}^+, [t_i, t_l] \Vdash \neg\psi$;
- $C_{[t_i, t_j]}$ contains exactly all formulae $\psi \in \text{CL}(\varphi)$ such that $\mathbf{M}^+, [t_i, t_j] \Vdash \psi$.

It is easy to check that, for every $[t_i, t_j] \in \mathbb{I}(\mathbb{T})^+$ and for every \mathcal{S} -successor t_{j+1} of t_j , $(R_{[t_i, t_j]}, C_{[t_i, t_j]})$ is an atom such that $(R_{[t_i, t_j]}, C_{[t_i, t_j]}) X_\varphi (R_{[t_i, t_{j+1}]}, C_{[t_i, t_{j+1}]})$.

For every $t_j \in T$, let $\mathbf{n}_j = \{(R_{[t_i, t_j]}, C_{[t_i, t_j]}) : t_i \mathbf{n} t_j\}$, $\mathcal{N} = \{\mathbf{n}_j : t_j \in T\}$, and $\mathcal{S} = \mathcal{S}_\varphi \cap (\mathcal{N} \times \mathcal{N})$. It is easy to check that for all $t_j \neq t_0$, \mathbf{n}_j is a node and that $\langle \mathcal{N}, \mathcal{S} \rangle$ is a fulfilling substructure. \square

The next theorem shows that a model for φ can be obtained by unfolding a fulfilling substructure $\langle \mathcal{N}, \mathcal{S} \rangle$, starting from its initial node \mathbf{n}_0 .

Theorem 4.10. *If there exists a fulfilling substructure $\langle \mathcal{N}, \mathcal{S} \rangle \subseteq \mathcal{T}_\varphi$, then the formula φ is satisfiable.*

Proof. Let $\langle \mathcal{N}, \mathcal{S} \rangle$ be a fulfilling substructure. To define a model for φ , we first build an infinite tree $\mathbb{T} = \langle T, \mathcal{S} \rangle$ by *unfolding* the fulfilling substructure $\langle \mathcal{N}, \mathcal{S} \rangle$ from its initial node \mathbf{n}_0 as follows:

- T is the (infinite) set of all finite non-empty \mathcal{S} -sequences $\mathbf{n}_0 \mathbf{n}_1 \dots \mathbf{n}_k$ that starts from the initial node \mathbf{n}_0 ;
- \mathcal{S} is such that, for any pair of points $t, t' \in T$, $(t, t') \in \mathcal{S}$ if and only if $t = \mathbf{n}_0 \dots \mathbf{n}_k$ and $t' = \mathbf{n}_0 \dots \mathbf{n}_k \mathbf{n}_{k+1}$ with \mathbf{n}_{k+1} \mathcal{S} -successor of \mathbf{n}_k .

Given an arbitrary order of the nodes of \mathcal{N} , we define a *total* order \triangleleft over finite \mathcal{S} -sequences, that is, on points of T , as follows:

- given two \mathcal{S} -sequences t and t' such that the length of t is less than the length of t' , we have that $t \triangleleft t'$;
- given two \mathcal{S} -sequences t and t' of the same length, $t \triangleleft t'$ if and only if t precedes t' on the lexicographical order based on the given (arbitrary) order of nodes in \mathcal{N} .

In order to build a model for φ , we define a suitable (partial) labelling function $L : \mathbb{I}(\mathbb{T})^+ \rightarrow \mathcal{A}_\varphi$. For any $t_j = \mathbf{n}_0 \dots \mathbf{n}_k$, L associates an atom $(R, C) \in \mathbf{n}_k$ with any interval $[t_i, t_j]$. We define L by (infinite) induction on the total order \triangleleft .

Base case. We start by defining the labelling of the initial interval $[t_0, t_0]$ (where $t_0 = \mathbf{n}_0$ is the root of \mathbb{T}). Since \mathbf{n}_0 is the initial node of $\langle \mathcal{N}, \mathcal{S} \rangle$, we have that $\mathbf{n}_0 = \langle (R_0, C_0), \emptyset \rangle$. We put $L([t_0, t_0]) = (R_0, C_0)$. Two cases may arise.

C_0 contains no existential formulae. In such a case, we define the labelling of an infinite branch starting from t_0 (remind that we admit only infinite models). Since $\langle \mathcal{N}, \mathcal{S} \rangle$ is a substructure, there exists a fulfilling path $\rho = \mathbf{n}_0 \mathbf{n}_1 \mathbf{n}_2 \dots$ starting from \mathbf{n}_0 .

Let $\sigma = t_0 t_1 t_2 \dots$ be the corresponding infinite branch in \mathbb{T} (where, for every $i \geq 0$, $t_i = \mathbf{n}_0 \dots \mathbf{n}_i$). For every $j \geq 0$, let $\mathbf{n}_j = \langle (R_{\mathbf{n}_j}, C_{\mathbf{n}_j}), M_{\mathbf{n}_j} \rangle$. We define the labelling $L([t_i, t_j])$ of every interval $[t_i, t_j]$, with $t_i, t_j \in \sigma$, in such a way that:

- for all $j > 0$, $L([t_j, t_j]) = (R_{\mathbf{n}_j}, C_{\mathbf{n}_j})$;
- for all $j > 0$ and all intervals $[t_i, t_j]$, with $t_i < t_j$, $L([t_i, t_j])$ is an X_φ -descendant of $L([t_i, t_i])$ in ρ .
- for every interval $[t_i, t_j]$, with $L([t_i, t_j]) = (R', C')$, if there exists a formula $A \diamond_r \theta \in R'$, then there exists a point $t_k \geq t_j$ such that $L([t_i, t_k]) = (R'', C'')$ and $\theta \in C''$ (see Definition 4.6).

Definitions 4.4 and 4.6 guarantee that there exists a labelling with such properties.

C_0 contains at least one existential formula. In such a case, for every existential formula $\psi \in C_0$, we guarantee that ψ gets satisfied by properly labelling an infinite branch starting from t_0 . Two cases may arise (depending on the structure of ψ).

- $\psi = E \diamond_r \theta$. By condition F1, either $\theta \in C_0$ (and thus θ is satisfied over the interval $[t_0, t_0]$) or there exists a path $\mathbf{n}_0 \mathbf{n}_1 \dots \mathbf{n}_k$ and a corresponding atom path $(R_0, C_0)(R_1, C_1) \dots (R_k, C_k)$, such that $\theta \in C_k$. In the latter case, let $t_0 t_1 \dots t_k$ be the branch in \mathbb{T} corresponding to $\mathbf{n}_0 \mathbf{n}_1 \dots \mathbf{n}_k$. We put, for every $i \leq k$, $L([t_0, t_i]) = (R_i, C_i)$, in order to satisfy θ over $[t_0, t_k]$. In both cases, we extend the finite branch to an infinite one and we define the labelling of all other intervals on the branch as in the case in which C_0 contains no existential formulae.
- $\psi = E \square_r \theta$. By condition F2, there exist a fulfilling infinite path $\mathbf{n}_0 \mathbf{n}_1 \dots$ and a corresponding infinite atom path $(R_0, C_0)(R_1, C_1) \dots$, such that $E \square_r \theta \in R_i$ for every $i \geq 0$. Let $t_0 t_1 \dots$ be the infinite branch in \mathbb{T} corresponding to $\mathbf{n}_0 \mathbf{n}_1 \dots$. We put, for every $i \geq 0$, $L([t_0, t_i]) = (R_i, C_i)$, in order to satisfy θ over $[t_0, t_i]$. As before, we define the labelling of all other intervals on the branch as in the case in which C_0 contains no existential formulae.

We repeat such a procedure for every existential formula in C_0 .

Inductive step. Let $t \in T$ such that (i) L is defined over all intervals $[t', t]$; (ii) for all $t' \triangleleft t$, either t' has been already taken into consideration or L is not defined over any interval $[t'', t']$; (iii) t has not been taken into consideration yet.

Consider the set $\text{REQ}(\mathbf{n})$, with \mathbf{n} such that $t = \mathbf{n}_0 \dots \mathbf{n}$, and suppose there exists an existential formula $\psi \in \text{REQ}(\mathbf{n})$. Two cases may arise. Either ψ is satisfied by the current labelling L , and we are done, or there are no branches starting from t that satisfy ψ . In the latter case, we satisfy ψ by defining a suitable labelling of an unlabeled infinite branch starting from t , as we have done in the base case of the induction. By repeating such a procedure for every existential formula $\psi \in \text{REQ}(\mathbf{n})$, we guarantee that all existential formulae in $\text{REQ}(\mathbf{n})$ are satisfied.

The model satisfying φ contains all and only the labelled (infinite) branches of \mathbb{T} . Let \mathbb{T}' be the infinite tree obtained from \mathbb{T} by removing all unlabeled branches and let \mathcal{V} be a valuation function \mathcal{V} such that, for every $p \in AP$ and $[t_i, t_j] \in \mathbb{I}(\mathbb{T}')^+$,

$[t_i, t_j] \in \mathcal{V}(p)$ if and only if $L([t_i, t_j]) = (R_{[t_i, t_j]}, C_{[t_i, t_j]})$ and $p \in C_{[t_i, t_j]}$. Induction on the structure of the formula that for every $\psi \in \text{CL}(\varphi)$ and for every $[t_i, t_j] \in \mathbb{I}(\mathbb{T}')^+$, we have that $\mathbf{M}^+, [t_i, t_j] \Vdash \psi$ if and only if $L([t_i, t_j]) = (R_{[t_i, t_j]}, C_{[t_i, t_j]})$ and $\psi \in C_{[t_i, t_j]}$.

- The base case, as well as the case of the propositional connectives \neg and \vee , are straightforward.
- Let ψ be the formula $E\Diamond_r\chi$. Suppose that $E\Diamond_r\chi \in C_{[t_i, t_j]}$. By the definition of L , there exists an infinite path $t_j t_{j+1} \dots$ and an interval $[t_j, t_k]$ such that $\chi \in C_{[t_j, t_k]}$. By inductive hypothesis, we have that $\mathbf{M}^+, [t_j, t_k] \Vdash \chi$, and thus $\mathbf{M}^+, [t_i, t_j] \Vdash E\Diamond_r\chi$.
As for the opposite implication, assume by contradiction that $\mathbf{M}^+, [t_i, t_j] \Vdash E\Diamond_r\chi$ and $E\Diamond_r\chi \notin C_{[t_i, t_j]}$. By atom definition, this implies that $\neg E\Diamond_r\chi = A\Box_r\neg\chi \in C_{[t_i, t_j]}$. By the definition of L , we have that $A\Box_r\neg\chi \in R_{[t_j, t_k]}$ for every $t_k \geq t_j$, and thus $\neg\chi \in C_{[t_j, t_k]}$. By inductive hypothesis, this implies that $\mathbf{M}^+, [t_j, t_k] \Vdash \neg\chi$ for every $t_k \geq t_j$, and thus $\mathbf{M}^+, [t_i, t_j] \Vdash A\Box_r\neg\chi$, which contradicts the hypothesis that $\mathbf{M}^+, [t_i, t_j] \Vdash E\Diamond_r\chi$.
- Let ψ be the formula $E\Box_r\chi$. Suppose that $E\Box_r\chi \in C_{[t_i, t_j]}$. By the definition of L , there exists an infinite path $\rho = t_j t_{j+1} \dots$ such that, for every $t_k \in \rho$, $E\Box_r\chi \in R_{[t_j, t_k]}$. By atom definition, this implies that $\chi \in C_{[t_j, t_k]}$ and, by inductive hypothesis, we have that $\mathbf{M}^+, [t_j, t_k] \Vdash \chi$, for every $t_k \in \rho$, $t_k \geq t_j$, and thus $\mathbf{M}^+, [t_i, t_j] \Vdash E\Box_r\chi$.
As for the opposite implication, assume by contradiction that $\mathbf{M}^+, [t_i, t_j] \Vdash E\Box_r\chi$ and $E\Box_r\chi \notin C_{[t_i, t_j]}$. By atom definition, this implies that $\neg E\Box_r\chi = A\Diamond_r\neg\chi \in C_{[t_i, t_j]}$. By the definition of L , we have that, for every infinite path $t_j t_{j+1} \dots$ starting from t_j , there exists $t_k \in \rho$, $t_k \geq t_j$ such that $\neg\chi \in C_{[t_j, t_k]}$. By inductive hypothesis, this implies that for every infinite path $\rho = t_j t_{j+1} \dots$ there exists a point $t_k \in \rho$, $t_k \geq t_j$, such that $\mathbf{M}^+, [t_j, t_k] \Vdash \neg\chi$, and thus $\mathbf{M}^+, [t_i, t_j] \Vdash A\Diamond_r\neg\chi$, which contradicts the hypothesis that $\mathbf{M}^+, [t_i, t_j] \Vdash E\Box_r\chi$.

Since $\langle \mathcal{N}, R \rangle$ is a substructure, $E\Diamond_r\varphi \in C_{[t_0, t_0]}$, and thus \mathbf{M}^+ is a model for φ . \square

4.3 The decision procedure

In this section, we present a decision procedure for BTNL, that progressively removes from \mathcal{T}_φ nodes that cannot contribute to fulfilling substructures.

Algorithm 1. Let φ be the formula we want to test for satisfiability. The decision procedure works as follows.

1. Build the (unique) initial tableau $\mathcal{T}_\varphi = \langle \mathcal{N}_\varphi, \mathcal{S}_\varphi \rangle$.
2. Look for a fulfilling substructure by repeatedly applying the following deletion rules, until no more nodes in the tableau can be deleted:
 - delete any node which is not \mathcal{S}_φ -reachable from an initial node;
 - delete any node such that there are no fulfilling paths starting from it;
 - delete any node which does not satisfy the conditions of Definition 4.8.

3. Let $\mathcal{T}^* = \langle \mathcal{N}^*, \mathcal{S}^* \rangle$ be the final tableau. If \mathcal{T}^* is not empty, return *true*, otherwise return *false*.

The check for the existence of fulfilling paths can be performed as follows. Given a formula $A \diamond_r \psi \in \text{CL}(\varphi)$, we execute the following marking procedure. First, for all nodes \mathbf{n} , mark all atoms $(R, C) \in \mathbf{n}$ such that $A \diamond_r \psi \in R$ and $\psi \in C$. Then, for all nodes \mathbf{n} , mark all unmarked atoms $(R, C) \in \mathbf{n}$ such that there exists an \mathcal{S}_φ -successor \mathbf{m} of \mathbf{n} that contains a marked atom (R', C') such that $(R, C) X_\varphi (R', C')$. Repeat this last step until no more atoms can be marked. Then, delete all nodes that either contain an unmarked atom (R, C) with $A \diamond_r \psi \in R$ or have no \mathcal{S}_φ -successors.

The other non trivial step of the algorithm is the removal of nodes that do not satisfy the conditions of Definition 4.8. Given a node \mathbf{n} , condition F1 can be easily checked by visiting the \mathcal{S}_φ -descendants of \mathbf{n} . Given an atom $(R_0, C_0) \in \mathbf{n}$, and a formula $E \square_r \psi \in R_0$, condition F2 is satisfied if we can find a *finite* path of nodes $\mathbf{n} = \mathbf{n}_0 \mathbf{n}_1 \dots \mathbf{n}_j \mathbf{n}_{j+1} \dots \mathbf{n}_k$ and a corresponding path of atoms $(R_0, C_0)(R_1, C_1) \dots (R_j, C_j) (R_{j+1}, C_{j+1}) \dots (R_k, C_k)$ such that

(i) $\mathbf{n}_j = \mathbf{n}_k$, (ii) $(R_j, C_j) = (R_k, C_k)$, (iii) for all $0 \leq i \leq k$, $E \square_r \psi \in R_i$, (iv) for every formula $A \diamond_r \theta \in R_0$, there exists $i \geq 0$ such that $\theta \in C_i$.

Furthermore, to guarantee that the infinite path

$$\mathbf{n}_0 \mathbf{n}_1 \dots \mathbf{n}_j \mathbf{n}_{j+1} \dots \mathbf{n}_{k-1} \mathbf{n}_j \mathbf{n}_{j+1} \dots \mathbf{n}_{k-1} \dots$$

is a fulfilling one, it suffices to check that, for every atom $(R', C') \in \mathbf{n}_j$ and every formula $A \diamond_r \xi \in C'$, either $\xi \in C'$ or there exists a node \mathbf{n}_l , with $j < l < k$, and an atom $(R'', C'') \in \mathbf{n}_l$ such that $\xi \in C''$ and (R'', C'') is a X_φ -descendant of (R', C') .

As for complexity issues, we have that:

- $|\mathcal{T}_\varphi| = 2^{2^{O(|\varphi|)}}$;
- all tests of step (ii) of the algorithm can be done in time polynomial in the size of $|\mathcal{T}_\varphi|$;
- after deleting at most $|\mathcal{N}_\varphi|$ nodes, the algorithm terminates.

Hence, checking the satisfiability for a BTNL formula has an overall time bound of $2^{2^{O(|\varphi|)}}$, that is, doubly exponential in the size of φ .

4.4 The decision procedure at work

In this section we apply the proposed decision procedure to the (satisfiable) formula $\varphi = E \square_r p$. We show only a portion of the whole tableau, which is sufficiently large to include a fulfilling substructure, and thus to prove that φ is satisfiable.

When searching for a fulfilling substructure for φ , we must take into consideration atoms which have been obtained by suitably combining one set of active requests with one set of current formulae. The non-empty sets of active requests are the following

ones:

$$\begin{aligned}
R_0 &= \{E\Diamond_r E\Box_r p, E\Box_r p\} & R_4 &= \{E\Diamond_r E\Box_r p\} \\
R_1 &= \{E\Diamond_r E\Box_r p, A\Diamond_r \neg p\} & R_5 &= \{A\Box_r A\Diamond_r \neg p\} \\
R_2 &= \{A\Box_r A\Diamond_r \neg p, E\Box_r p\} & R_6 &= \{E\Box_r p\} \\
R_3 &= \{A\Box_r A\Box_r \neg p, A\Diamond_r \neg p\} & R_7 &= \{A\Diamond_r \neg p\}
\end{aligned}$$

while the sets of current formulae are the following ones:

$$\begin{aligned}
C_0 &= \{E\Diamond_r E\Box_r p, E\Box_r p, p\} & C_4 &= \{A\Box_r A\Diamond_r \neg p, E\Box_r p, p\} \\
C_1 &= \{E\Diamond_r E\Box_r p, E\Box_r p, \neg p\} & C_5 &= \{A\Box_r A\Diamond_r \neg p, E\Box_r p, \neg p\} \\
C_2 &= \{E\Diamond_r E\Box_r p, A\Diamond_r \neg p, p\} & C_6 &= \{A\Box_r A\Diamond_r \neg p, A\Diamond_r \neg p, p\} \\
C_3 &= \{E\Diamond_r E\Box_r p, A\Diamond_r \neg p, \neg p\} & C_7 &= \{A\Box_r A\Diamond_r \neg p, A\Diamond_r \neg p, \neg p\}
\end{aligned}$$

As an example, consider the initial node $\mathbf{n}_0 = \langle (R_0, C_0), \emptyset \rangle$. Figure 4.2 depicts a portion of \mathcal{T}_φ that is \mathcal{S}_φ -reachable from \mathbf{n}_0 . In every node \mathbf{n} the atom above the line is the designated atom $(R_{\mathbf{n}}, C_{\mathbf{n}})$, while the atoms below the line are the atoms in $M_{\mathbf{n}}$.

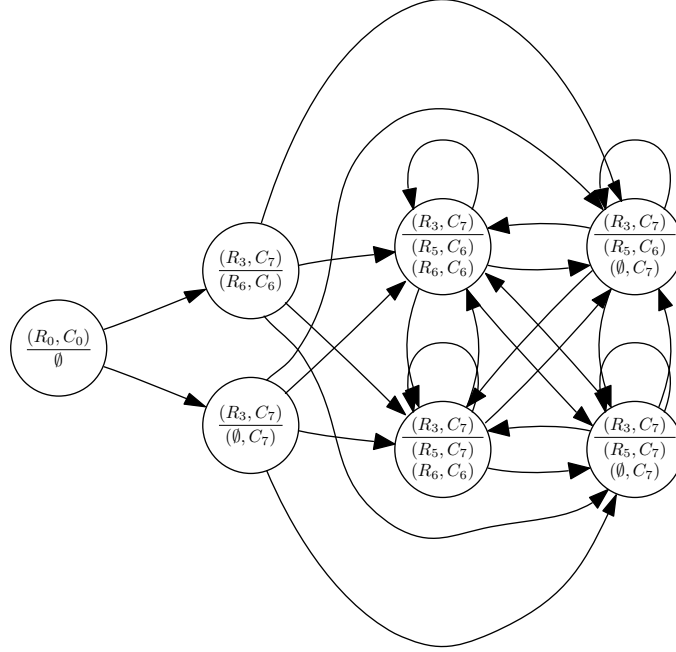


Figure 4.2: A portion of the tableau for $E\Box_r p$.

In the considered portion of \mathcal{T}_φ the only atom with $A\Diamond_r$ -formulae in its set of active requests is (R_3, C_7) , since $A\Diamond_r \neg p \in R_3$. The atom (R_3, C_7) immediately fulfills $A\Diamond_r \neg p$, since $\neg p \in C_7$. Hence, for every node of the substructure of Figure 4.2 there exists a fulfilling path. The only atom containing a formula of the form $E\Diamond_r \psi$ is (R_0, C_0) , since $E\Diamond_r E\Box_r p \in R_0$. (R_0, C_0) immediately fulfills $E\Diamond_r E\Box_r p$, and thus condition F1 is trivially satisfied. As for condition F2, consider the atom (R_6, C_6) .

It is easy to see that, for every node \mathbf{n} containing (R_6, C_6) , there exists a fulfilling infinite path starting from \mathbf{n} such that every node contains (R_6, C_6) . Thus, condition F2 is satisfied. This allows us to conclude that the substructure depicted in Figure 4.2 is fulfilling, and thus our decision procedure correctly concludes that the formula $E\Box_r p$ is satisfiable.

5

The tableau method for PNL

In Chapter 3 we have established the decidability of RPNL over the natural numbers. We basically proved that an RPNL formula is satisfiable if and only if there exists a finite model, or an ultimately periodic (infinite) one, with a finite representation of *bounded size*. In both cases, such a model can be built starting from any model satisfying the formula by progressively removing exceeding points from it until the desired bound is reached. The removal of a point d from a model causes the removal of all intervals either beginning or ending at it. Since RPNL features only future time modalities, the removal of intervals beginning at d is not critical. By contrast, the removal of intervals ending at d may introduce “defects”, that is, there may be existential future temporal formulae that are not satisfied any more. However, by properly choosing the point d to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects (by possibly changing the truth value of formulas over intervals ending at them) without introducing new defects.

In this chapter, we generalize the technique we used for RPNL to full PNL by showing that a PNL formula is satisfiable if and only if there exists a finite model, or an infinite one, with a finite representation of *bounded size*. As in the case of RPNL, such a model can be obtained by removing exceeding points from a given model satisfying the formula, but the removal process turns out to be much more involved. In contrast with the case of RPNL, the removal of a point d from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of d . Hence, to fix the defects possibly caused by the removal of d , we must guarantee that there exist sufficiently many points with the same characteristics as d both in the future and in the past of d . Moreover, we must be sure that changing the valuation of intervals that either end or start at these points does not generate new defects. In the following, we show that this can actually be done. This chapter is an extended version of [BMS07a].

5.1 Labelled Interval Structures for PNL

In this section we introduce some preliminary notions and we establish some basic results on which our tableau method for PNL relies. As in the previous chapter,

we first consider the case of the most general variant of PNL, $\text{PNL}^{\pi+}$. Then, in Section 5.3, we will show how the proposed method can be adapted to PNL^+ and PNL^- . Furthermore, we restrict our attention to the case of the integers. From now on, all linear orderings we will consider are isomorphic to \mathbb{Z} (with the usual ordering) or to a subset of it.

Let φ be a $\text{PNL}^{\pi+}$ formula to be checked for satisfiability and let AP be the set of its propositional letters.

Definition 5.1. The *closure* $\text{CL}(\varphi)$ of φ is the set of all subformulae of $\diamond_r\varphi$ and of their negations (we identify $\neg\neg\psi$ with ψ).

As will become clear later, we put the formula $\diamond_r\varphi$ and its negation in $\text{CL}(\varphi)$ to guarantee the existence of at least one interval over which φ holds.

Definition 5.2. The set of *temporal formulae* of φ is the set $\text{TF}(\varphi) = \{\diamond_r\psi, \square_r\psi, \diamond_l\psi, \square_l\psi \in \text{CL}(\varphi)\}$.

By induction on the structure of φ , we can easily prove that, for every formula φ , $|\text{CL}(\varphi)|$ is less than or equal to $2 \cdot (|\varphi| + 1)$, while $|\text{TF}(\varphi)|$ is less than or equal to $2 \cdot |\varphi|$. We are now ready to introduce the notion of φ -atom.

Definition 5.3. A φ -atom is a set $A \subseteq \text{CL}(\varphi)$ such that:

- for every $\psi \in \text{CL}(\varphi)$, $\psi \in A$ iff $\neg\psi \notin A$;
- for every $\psi_1 \vee \psi_2 \in \text{CL}(\varphi)$, $\psi_1 \vee \psi_2 \in A$ iff $\psi_1 \in A$ or $\psi_2 \in A$.

We denote the set of all φ -atoms by \mathcal{A}_φ . We have that $|\mathcal{A}_\varphi| \leq 2^{|\varphi|+1}$. Atoms are connected by the following binary relation.

Definition 5.4. Let LR_φ be a relation such that for every pair of atoms $A_1, A_2 \in \mathcal{A}_\varphi$, $A_1 LR_\varphi A_2$ if and only if (i) for every $\square_r\psi \in \text{CL}(\varphi)$, if $\square_r\psi \in A_1$ then $\psi \in A_2$ and (ii) for every $\square_l\psi \in \text{CL}(\varphi)$, if $\square_l\psi \in A_2$ then $\psi \in A_1$.

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

Definition 5.5. A φ -labeled interval structure (LIS for short) is a tuple $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+ \rangle$ is a non-strict interval structure and $\mathcal{L} : \mathbb{I}(\mathbb{D})^+ \rightarrow \mathcal{A}_\varphi$ is a *labeling function* such that, (a) for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, $\pi \in \mathcal{L}([d_i, d_j])$ iff $d_i = d_j$, and (b) for every pair of neighboring intervals $[d_i, d_j], [d_j, d_k] \in \mathbb{I}(\mathbb{D})^+$, $\mathcal{L}([d_i, d_j]) LR_\varphi \mathcal{L}([d_j, d_k])$.

If we interpret the labeling function as a valuation function, LISs represent *candidate models* for φ . The truth of formulae devoid of temporal operators, that of the modal constant πm and that of \square_r/\square_l formulae indeed follow from the definition of φ -atom and LR_φ , respectively. However, to obtain a model for φ , we must also guarantee the truth of \diamond_r/\diamond_l formulae. To this end, we introduce the notion of fulfilling LIS.

Definition 5.6. A φ -labelled interval structure $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ is *fulfilling* if and only if (a) for every temporal formula $\diamond_r \psi \in \text{TF}(\varphi)$ and every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, if $\diamond_r \psi \in \mathcal{L}([d_i, d_j])$, then there exists $d_k \geq d_j$ such that $\psi \in \mathcal{L}([d_j, d_k])$ and (b) for every temporal formula $\diamond_l \psi \in \text{TF}(\varphi)$ and every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$, if $\diamond_l \psi \in \mathcal{L}([d_i, d_j])$, then there exists $d_k \leq d_i$ such that $\psi \in \mathcal{L}([d_k, d_i])$.

The next theorem proves that for any given formula φ , the satisfiability of φ is equivalent to the existence of a fulfilling LIS with an interval labeled by φ .

Theorem 5.7. *A formula φ is satisfiable if and only if there exists a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ with $\varphi \in \mathcal{L}([d_i, d_j])$ for some $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$.*

The implication from left to right is straightforward; the opposite implication is proved by induction on the structure of the formula.

From now on, we say that a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ *satisfies* φ if and only if there exists an interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^+$ such that $\varphi \in \mathcal{L}([d_i, d_j])$. 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.

Definition 5.8. Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ and a point $d \in D$, we define the set of *future temporal requests* of d as the set $\text{REQ}_f^{\mathbf{L}}(d) = \{\diamond_r \xi / \square_r \xi \in \text{TF}(\varphi) : \exists d' \in D(\diamond_r \xi / \square_r \xi \in \mathcal{L}([d', d]))\}$ and the set of *past temporal requests* of d as the set $\text{REQ}_p^{\mathbf{L}}(d) = \{\diamond_l \xi / \square_l \xi \in \text{TF}(\varphi) : \exists d' \in D(\diamond_l \xi / \square_l \xi \in \mathcal{L}([d, d']))\}$. The set of *temporal requests* of d is defined as $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}_p^{\mathbf{L}}(d) \cup \text{REQ}_f^{\mathbf{L}}(d)$.

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

Definition 5.9. Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$, $D' \subseteq D$, and $\mathcal{R} \in \text{REQ}_\varphi$, we say that \mathcal{R} *occurs n times in D'* if and only if there exist exactly n distinct points $d_{i_1}, \dots, d_{i_n} \in D'$ such that $\text{REQ}^{\mathbf{L}}(d_{i_j}) = \mathcal{R}$, for all $1 \leq j \leq n$.

We describe the process of removing a point from a LIS. Given $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ and $d \in D$, let \mathbf{L}_{-d} be the set of all LIS $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L}' \rangle$ such that $D' = D \setminus \{d\}$ and $\text{REQ}^{\mathbf{L}'}(\bar{d}) = \text{REQ}^{\mathbf{L}}(\bar{d})$, for all $\bar{d} \in D \setminus \{d\}$. \mathbf{L} and \mathbf{L}' do not necessarily agree on the labeling of intervals, but they agree on the sets of requests of points.

Given a fulfilling LIS \mathbf{L} and a point d , it is not guaranteed that \mathbf{L}_{-d} contains a fulfilling LIS. The removal of d indeed causes the removal of all intervals either beginning or ending at it and thus there can be a point $\bar{d} < d$ (resp., $\bar{d} > d$) such that there exists a formula $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(\bar{d})$ (resp., $\diamond_l \psi \in \text{REQ}_p^{\mathbf{L}}(\bar{d})$) which is fulfilled in \mathbf{L} , but not in any $\mathbf{L}' \in \mathbf{L}_{-d}$. The following lemma provides a sufficient condition for preserving the fulfilling property when removing a point from \mathbf{L} .

Lemma 5.10. *Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS, f be the number of \diamond_r -formulae in $\text{TF}(\varphi)$, and p be the number of \diamond_l -formulae in $\text{TF}(\varphi)$. If there exists a*

point $d_e \in D$ such that (i) there exist at least $f \cdot p + p$ distinct points $d < d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$ and (ii) there exist at least $f \cdot p + f$ distinct points $d > d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$, then there is one fulfilling LIS $\widehat{\mathbf{L}} \in \mathbf{L}_{-d_e}$.

Proof. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS and let $d_e \in D$ be a point such that there exist at least $f \cdot p + p$ distinct points $d < d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$ and at least $f \cdot p + f$ distinct points $d > d_e$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_e)$. We define $\mathbb{D}' = \langle D \setminus \{d_e\}, < \rangle$ and $\mathcal{L}' = \mathcal{L}|_{\mathbb{I}(\mathbb{D}')^+}$ (the restriction of \mathcal{L} to the intervals on \mathbb{D}'). The pair $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L}' \rangle$ is obviously a LIS in \mathbf{L}_{-d_e} , but, as already pointed out, it is not necessarily a fulfilling one. We show how the defects possibly caused by the removal of d_e can be fixed one-by-one by properly redefining \mathcal{L}' .

Consider the case of a point $d < d_e$ and a formula $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(d)$ such that $\psi \in \mathcal{L}([d, d_e])$ and there are no $\bar{d} \in D \setminus \{d_e\}$ such that $\psi \in \mathcal{L}'([d, \bar{d}])$ (the symmetric case of $d > d_e$ and $\diamond_l \psi \in \text{REQ}_p^{\mathbf{L}}(d)$ can be dealt with in the same way). Let $R = \{d_r \in D : d_r > d_e \wedge \text{REQ}^{\mathbf{L}}(d_r) = \text{REQ}^{\mathbf{L}}(d_e)\}$. To satisfy the request $\diamond_r \psi \in \text{REQ}^{\mathbf{L}}(d)$ we change the labeling of an interval $[d, d_r]$, for a suitable $d_r \in R$. However, to prevent such a change from making one or more requests in $\text{REQ}_p^{\mathbf{L}}(d_r)$ no longer satisfied, we preliminarily redefine the labeling \mathcal{L}' . First, we take a minimal set of points $P^{d_e} \subseteq D \setminus \{d_e\}$ such that, for every $\diamond_l \psi \in \text{REQ}_p^{\mathbf{L}}(d_e)$ there exists a point $d_i \in P^{d_e}$ such that $\psi \in \mathcal{L}([d_i, d_e])$. We call P^{d_e} the set of *preserved past points* for d_e . Then, for every point $d_i \in P^{d_e}$, let $F^{d_i} \subseteq D$ be a minimal set of points such that, for every $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(d_i)$ there is a point $d_f \in F^{d_i}$ such that $\psi \in \mathcal{L}([d_i, d_f])$. We call F^{d_i} the set of *preserved future points* for d_i .

Let G be the set of points $R \setminus \bigcup_{d_i \in P^{d_e}} F^{d_i}$. By the minimality requirements, $|P^{d_e}|$ is bounded by p and $|F^{d_i}|$, for each $d_i \in P^{d_e}$, is bounded by f . Hence, $|\bigcup_{d_i \in P^{d_e}} F^{d_i}| \leq f \cdot p$ and, by Condition (ii), $|G|$ is greater than or equal to f . Now, we can use points in G to fulfill $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(d)$, without generating new defects, as follows. Since $\text{REQ}_f^{\mathbf{L}}(d)$ contains at most f \diamond_r -formulae, there exists at least one point $d_g \in G$ such that the atom $\mathcal{L}'([d, d_g])$ either fulfills no \diamond_r -formulae or it fulfills only \diamond_r -formulae which are also fulfilled by an φ -atom $\mathcal{L}'([d, d_k])$ for some d_k . Let d_g one of these “useless” points. We can redefine $\mathcal{L}'([d, d_g])$ by putting $\mathcal{L}'([d, d_g]) = \mathcal{L}([d, d_e])$, thus fixing the problem for $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(d)$. Since $\text{REQ}^{\mathbf{L}}(d_g) = \text{REQ}^{\mathbf{L}}(d_e)$, such a change has no impact on the right neighboring intervals of $[d, d_g]$. By contrast, there may exist one or more \diamond_l -formulae in $\text{REQ}_p^{\mathbf{L}}(d_g)$ which, due to the change in the labeling of $[d, d_g]$, are not satisfied anymore. In such a case, however, we can recover satisfiability, without introducing any new defect, by putting $\mathcal{L}'([d_i, d_g]) = \mathcal{L}([d_i, d_e])$ for all $d_i \in P^{d_e}$.

In the same way, we can fix all possible other defects caused by the removal of d_e . Let $\widehat{\mathbf{L}} = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \widehat{\mathcal{L}} \rangle$ be the resulting LIS. It is immediate that $\widehat{\mathbf{L}}$ is fulfilling and it belongs to \mathbf{L}_{-d_e} . \square

By taking advantage of Lemma 5.10, we can prove the following theorem.

Theorem 5.11. *Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a finite fulfilling LIS that satisfies φ , f be the number of \diamond_r -formulae in $\text{TF}(\varphi)$, and p be the number of \diamond_l -formulae in $\text{TF}(\varphi)$.*

Then, there exists a finite fulfilling LIS $\widehat{\mathbf{L}} = \langle \widehat{\mathbb{D}}, \mathbb{I}(\widehat{\mathbb{D}})^+, \widehat{\mathcal{L}} \rangle$ that satisfies φ such that, for every $\widehat{d}_i \in \widehat{D}$, $\text{REQ}^{\widehat{\mathbf{L}}}(\widehat{d}_i)$ occurs at most $m = 2fp + f + p$ times in \widehat{D} .

Proof. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a finite fulfilling LIS that satisfies φ . If for every $d_j \in D$, $\text{REQ}^{\mathbf{L}}(d_j)$ occurs at most m times in D , 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 $\mathcal{R}_0 = \{\text{REQ}_1, \text{REQ}_2, \dots, \text{REQ}_k\}$ be the (arbitrarily ordered) finite set of all and only the sets of requests that occur more than m times in D . \mathbf{L}_0 can be turned into a fulfilling LIS $\mathbf{L}_1 = \langle \mathbb{D}_1, \mathbb{I}(\mathbb{D}_1)^+, \mathcal{L}_1 \rangle$ satisfying φ , which contains exactly m points $d \in D_1$ such that $\text{REQ}^{\mathbf{L}_1}(d) = \text{REQ}_1$ as follows. Since REQ_1 occurs more than m times in D , there exists a point $d_e \in D$ such that $\text{REQ}^{\mathbf{L}_0}(d_e) = \text{REQ}_1$ and there exist at least $fp+p$ distinct points $d < d_e$ such that $\text{REQ}^{\mathbf{L}_0}(d) = \text{REQ}^{\mathbf{L}_0}(d_e)$ and at least $fp+f$ distinct points $d > d_e$ such that $\text{REQ}^{\mathbf{L}_0}(d) = \text{REQ}^{\mathbf{L}_0}(d_e)$. Hence, by Lemma 5.10, there exists a fulfilling LIS $\mathbf{L}' \in \mathbf{L}_{-d_e}$. We repeat the application of Lemma 5.10 until we get a fulfilling LIS \mathbf{L}_1 such that REQ_1 occurs exactly m times in D_1 . It remains to show that \mathbf{L}_1 satisfies φ . Since \mathbf{L}_0 satisfies φ , we have that there exists an interval $[d_i, d_j]$ such that $\varphi \in \mathcal{L}_0([d_i, d_j])$. By definition of $\text{CL}(\varphi)$, $\diamond_r \varphi \in \text{CL}(\varphi)$, hence $\diamond_r \varphi \in \text{REQ}^{\mathbf{L}_0}(d_i)$. In \mathbf{L}_1 two cases are possible: either $d_i \in D_1$ or it does not. If $d_i \in D_1$, then $\diamond_r \varphi \in \text{REQ}^{\mathbf{L}_1}(d_i)$ and, being \mathbf{L}_1 fulfilling, there exists an interval $[d_i, d_k]$ such that $\varphi \in \mathcal{L}_1([d_i, d_k])$. If $d_i \notin D_1$, then it has been deleted at some stage of the construction of \mathbf{L}_1 . This implies that $\text{REQ}^{\mathbf{L}_1}(d_i) = \text{REQ}_1$ and thus there exist m points d in \mathbf{L}_1 such that $\text{REQ}^{\mathbf{L}_1}(d) = \text{REQ}_1$. Since \mathbf{L}_1 is fulfilling, there exists an interval $[d, d']$ such that $\varphi \in \mathcal{L}_1([d, d'])$. In both cases \mathbf{L}_1 satisfies φ .

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 . \square

Let us consider now the case of infinite (fulfilling) LISs. We start with a classification of points belonging the domain of the structure.

Definition 5.12. Given an infinite LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$, we partition the points in D into the following sets:

- $\text{Fin}(\mathbf{L})$ is the set of all points $d \in D$ such that $\text{REQ}^{\mathbf{L}}(d)$ occurs finitely many times in D ;
- $\text{Inf}_l(\mathbf{L})$ is the set of all points $d \in D$ such that $\text{REQ}^{\mathbf{L}}(d)$ occurs infinitely many times in D , but there exists a point d_{max} such that, for all $d' > d_{max}$, $\text{REQ}^{\mathbf{L}}(d') \neq \text{REQ}^{\mathbf{L}}(d)$;
- $\text{Inf}_r(\mathbf{L})$ is the set of all points $d \in D$ such that $\text{REQ}^{\mathbf{L}}(d)$ occurs infinitely many times in D , but there exists a point d_{min} such that, for all $d' < d_{min}$, $\text{REQ}^{\mathbf{L}}(d') \neq \text{REQ}^{\mathbf{L}}(d)$;
- $\text{Inf}(\mathbf{L})$ is the set of all points $d \in D$ such that $\text{REQ}^{\mathbf{L}}(d)$ occurs infinitely many times in D and, for every point d' , there exists $d'' < d'$ such that $\text{REQ}^{\mathbf{L}}(d'') = \text{REQ}^{\mathbf{L}}(d)$ and there exists $d''' > d'$ such that $\text{REQ}^{\mathbf{L}}(d''') = \text{REQ}^{\mathbf{L}}(d)$.

The following definition captures a particular subclass of infinite LISs that enjoy a finite representation.

Definition 5.13. An infinite LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ is *ultimately periodic*, with left period l , infix i and right period r , if and only if there exists $d_0 \in D$ such that for all $k < 0$, $\text{REQ}^{\mathbf{L}}(d_k) = \text{REQ}^{\mathbf{L}}(d_{k-l})$ and for all $k \geq 0$, $\text{REQ}^{\mathbf{L}}(d_{i+k}) = \text{REQ}^{\mathbf{L}}(d_{i+k+r})$.

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

Theorem 5.14. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be an infinite fulfilling LIS that satisfies φ , f be the number of \diamond_r -formulae in $\text{TF}(\varphi)$, and p be the number of \diamond_l -formulae in $\text{TF}(\varphi)$. Then, there exists an ultimately periodic fulfilling LIS $\widehat{\mathbf{L}} = \langle \widehat{\mathbb{D}}, \mathbb{I}(\widehat{\mathbb{D}})^+, \widehat{\mathcal{L}} \rangle$, with left period l , infix i and right period r , such that

1. for every $d_j \in \text{Fin}(\widehat{\mathbf{L}})$, $\text{REQ}^{\widehat{\mathbf{L}}}(d_j)$ occurs at most $m = 2fp + f + p$ times in D ;
2. for every $d_j \in \text{Inf}_r(\widehat{\mathbf{L}})$, $\text{REQ}^{\widehat{\mathbf{L}}}(d_j)$ occurs exactly $fp + p$ times in I , where I is the set of points in the infix part of $\widehat{\mathbf{L}}$;
3. for every $d_j \in \text{Inf}_l(\widehat{\mathbf{L}})$, $\text{REQ}^{\widehat{\mathbf{L}}}(d_j)$ occurs exactly $fp + f$ times in I ;
4. for all points $d_j \in \text{Inf}(\widehat{\mathbf{L}})$, $d_j \notin I$;
5. $r \leq |\text{REQ}_\varphi|$ and $l \leq |\text{REQ}_\varphi|$;
6. for every $d_j \in \text{Fin}(\mathbf{L})$ and every formula $\diamond_r \psi \in \text{REQ}_f^{\mathbf{L}}(d_j)$, there exists a point $d_h \leq d_{i+(f \cdot p + f) \cdot r}$ such that $\psi \in \widehat{\mathcal{L}}([d_j, d_h])$;
7. for every $d_j \in \text{Fin}(\mathbf{L})$ and every formula $\diamond_l \psi \in \text{REQ}_f^{\mathbf{L}}(d_j)$, there exists a point $d_h \geq d_{-(f \cdot p + p) \cdot l}$ such that $\psi \in \widehat{\mathcal{L}}([d_h, d_j])$

that satisfies φ .

Proof. Let φ be a satisfiable formula and let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be an infinite fulfilling LIS that satisfies φ . By exploiting Lemma 5.10, we briefly show how to build a fulfilling LIS $\widehat{\mathbf{L}}$ which respects Conditions 1–7.

1. Let d_0 be the smallest point in $\text{Fin}(\mathbf{L}) \cup \text{Inf}_r(\mathbf{L})$ and d_{i-1} be the greatest point in $\text{Fin}(\mathbf{L}) \cup \text{Inf}_l(\mathbf{L})$. The set $I = \{d_0, \dots, d_{i-1}\}$ will be the infix of $\widehat{\mathbf{L}}$. By repeatedly applying Lemma 5.10 we can remove from the infix all points $d \in \text{Fin}(\mathbf{L})$ such that $\text{REQ}^{\mathbf{L}}(d)$ occurs more than m times in D .
2. Suppose that there exists a point $d_j \in \text{Inf}_r(\mathbf{L})$ such that $\text{REQ}^{\mathbf{L}}(d_j)$ does not occur $fp + p$ times in I : two cases may arise. If $\text{REQ}^{\mathbf{L}}(d_j)$ occurs more than $fp + p$ times in I , we can exploit Lemma 5.10 to remove the exceeding occurrences of $\text{REQ}^{\mathbf{L}}(d_j)$. If $\text{REQ}^{\mathbf{L}}(d_j)$ occurs less than $fp + p$ times in I , let $d_k > d_{i-1}$ be the point such that $\text{REQ}^{\mathbf{L}}(d_k) = \text{REQ}^{\mathbf{L}}(d_j)$ and $\text{REQ}^{\mathbf{L}}(d_k)$ occurs *exactly* $fp + p$ times in $\{d_0, \dots, d_{i-1}, \dots, d_k\}$. $I = \{d_0, \dots, d_{i-1}, \dots, d_k\}$ becomes the new infix of the ultimately periodic LIS. We repeat such a procedure until $\text{REQ}^{\mathbf{L}}(d)$ occurs exactly $fp + p$ times in I , for all $d \in \text{Inf}_r(\mathbf{L})$.

3. Suppose that there exists a point $d_j \in \text{Inf}_l(\mathbf{L})$ such that $\text{REQ}^{\mathbf{L}}(d_j)$ does not occur $fp+f$ times in I . We proceed as in the previous case, either by removing the exceeding occurrences of $\text{REQ}^{\mathbf{L}}(d_j)$ or by extending the infix to the left if $\text{REQ}^{\mathbf{L}}(d_j)$ occurs less than $fp+f$ times in I .
4. Suppose now that there exists a point $d_j \in \text{Inf}(\mathbf{L})$ such that $d_j \in I$. By the definition of $\text{Inf}(\mathbf{L})$, there are infinitely many points $d < d_j$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_j)$ and infinitely many points $d > d_j$ such that $\text{REQ}^{\mathbf{L}}(d) = \text{REQ}^{\mathbf{L}}(d_j)$. Hence, by exploiting Lemma 5.10, we can obtain a fulfilling LIS satisfying φ where d_j is removed.
5. Let $I = \{d_0, \dots, d_{i-1}\}$ be the infix of \mathbf{L} and suppose that it respects Conditions 1–4. To turn \mathbf{L} into an ultimately periodic LIS respecting Condition 5, we must show how to define the right and left period. Consider the set $\mathcal{R} = \{\text{REQ}^{\mathbf{L}}(d) : d \in \text{Inf}(\mathbf{L}) \cup \text{Inf}_r(\mathbf{L})\}$ and let $\mathcal{R} = \{\text{REQ}_0, \dots, \text{REQ}_{r-1}\}$ be an arbitrary enumeration of it. The cardinality r of \mathcal{R} will be the right period of $\widehat{\mathbf{L}}$. We inductively define $\widehat{\mathbf{L}}$ in such a way that, for all $k \geq 0$, $\text{REQ}^{\widehat{\mathbf{L}}}(\widehat{d}_{i+k}) = \text{REQ}_{k \bmod r}$. Let $k = 0$, and consider $\text{REQ}^{\mathbf{L}}(d_i)$: if $\text{REQ}^{\mathbf{L}}(d_i) = \text{REQ}_0$, we are done. Otherwise, let $d_h > d_i$ be the first occurrence of REQ_0 after d_i . Since \mathbf{L} respects Conditions 1–4, we have that, for every point $d_i \leq d' < d_h$, there exist sufficiently many points $d'' < d_i$ such that $\text{REQ}^{\mathbf{L}}(d'') = \text{REQ}^{\mathbf{L}}(d')$. Hence, by Lemma 5.10, there exists a LIS \mathbf{L}_0 where all points $d_i \leq d' < d_h$ have been removed. Thus, \mathbf{L}_0 is such that $\text{REQ}^{\mathbf{L}_0}(d_i) = \text{REQ}^{\mathbf{L}}(d_h) = \text{REQ}_0$. Now, let $k > 0$ and suppose that $\mathbf{L}_{k-1} = \langle \mathbb{D}_{k-1}, \mathbb{I}(\mathbb{D}_{k-1}), \mathcal{L}_{k-1} \rangle$ respect the condition for all $h < k$. We can proceed as in the case of $k = 0$ and define a LIS $\mathbf{L}_k = \langle \mathbb{D}_k, \mathbb{I}(\mathbb{D}_k), \mathcal{L}_k \rangle$ such that $\text{REQ}^{\mathbf{L}_k}(d_{i+k}) = \text{REQ}_{k \bmod r}$.
The left period of $\widehat{\mathbf{L}}$ can be defined in an analogous way starting from an arbitrary enumeration of the set $\mathcal{L} = \{\text{REQ}^{\mathbf{L}}(d) : d \in \text{Inf}(\mathbf{L}) \cup \text{Inf}_l(\mathbf{L})\}$.
6. Suppose that \mathbf{L} respects Conditions 1-5. Let $d_j \in \text{Fin}(\mathbf{L})$ and $\diamond_r \psi \in \text{REQ}^{\mathbf{L}}(d_j)$ be a formula that is fulfilled only by intervals $[d_j, d_h]$ such that $d_h > d_{i+f(p+1)r}$. Since \mathbf{L} respects Condition 2, for every point d' such that $d_i \leq d' < d_h$ we have that there exist at least $fp+p$ points $d'' < d_i$ with $\text{REQ}^{\mathbf{L}}(d'') = \text{REQ}^{\mathbf{L}}(d')$. Hence, we can exploit Lemma 5.10 to remove points between d_i and d_h , thus building a fulfilling LIS $\widehat{\mathbf{L}}$ that satisfies φ and such that $\diamond_r \psi \in \text{REQ}^{\widehat{\mathbf{L}}}(d_j)$ is fulfilled by an interval $[d_j, d_h]$ with $d_h \leq d_{i+(fp+f)r}$.
7. To build a fulfilling LIS $\widehat{\mathbf{L}}$ that respects Condition 7, we suppose that \mathbf{L} respects Conditions 1-5, and we proceed with a removal procedure analogous to the one for the previous case. \square

5.2 A tableau-based decision procedure for PNL^{π+}

In this section we define a tableau method for PNL^{π+} over the integers or a subset of them, that resembles the one we developed for RPNL interpreted over the naturals in the previous chapter.

Given a formula φ , let $m = 2fp + f + p$, where f (resp. p) is the number

of \diamond_r -formulae (resp. \diamond_l -formulae) in $\text{CL}(\varphi)$. A tableau for $\text{PNL}^{\pi+}$ is a special *decorated tree* \mathcal{T} . For each node n in a branch B , the *decoration* $\nu(n)$ is a tuple $\langle [d_i, d_j], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, where:

- $[d_i, d_j] \in \mathbb{I}(\mathbb{D}_n)^+$;
- $\text{REQ}_n : D_n \mapsto \text{REQ}_\varphi$ is a *request function*;
- $\mathbb{D}_n = \langle D_n, < \rangle$ is a finite linear order;
- $A_n \in \mathcal{A}_\varphi$ is such that: (i) $\pi \in A_n$ if and only if $d_i = d_j$, (ii) for all $\square_r \psi \in \text{REQ}_n(d_i)$, $\psi \in A_n$, (iii) for all $\square_l \psi \in \text{REQ}_n(d_j)$, $\psi \in A_n$, (iv) for all $\psi \in A_n$, if $\psi = \diamond_l \xi$ or $\psi = \square_l \xi$, then $\psi \in \text{REQ}_n(d_i)$, and (v) for all $\psi \in A_n$, if $\psi = \diamond_r \xi$ or $\psi = \square_r \xi$, then $\psi \in \text{REQ}_n(d_j)$;

The root r of the tree is decorated by the *empty decoration* $\langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$.

Given a node $n \in B$, decorated with $\langle [d_i, d_j], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, and a future existential formula $\diamond_r \psi \in A_n$, we say that $\diamond_r \psi$ is *fulfilled on B* if and only if there exists a node $n' \in B$ such that $\nu(n') = \langle [d_j, d_k], A_{n'}, \text{REQ}_{n'}, \mathbb{D}_{n'} \rangle$ and $\psi \in A_{n'}$. Conversely, we say that a past existential formula $\diamond_l \psi \in A_n$ is *fulfilled on B* if and only if there exists a node $n' \in B$ such that $\nu(n') = \langle [d_k, d_i], A_{n'}, \text{REQ}_{n'}, \mathbb{D}_{n'} \rangle$ and $\psi \in A_{n'}$. A node n is said to be *active on B* if and only if A_n contains at least one (future or past) existential formula which is not fulfilled on B .

Expansion rules. Let B a branch of a decorated tree \mathcal{T} . We denote by \mathbb{D}_B and REQ_B the linear order and the request function of the decoration of the last node in B , respectively. Moreover, let d_l and d_r be the minimum and maximum element of \mathbb{D}_B , respectively. The *expansion rules* for B are:

1. *Right step rule:* if there exists an active node $n \in B$, with $\nu(n) = \langle [d_i, d_j], A_n, \text{REQ}_n, \mathbb{D}_n, x \rangle$ and a non-fulfilled future existential formula in A_n , then extend D_B to $D' = D_B \cup \{d_{r+1}\}$, with $d_{r+1} > d_r$. Then, take an atom A' such that $A_n \text{ LR}_\varphi A'$ and extend REQ_B to $\text{REQ}' : D' \mapsto \text{REQ}_\varphi$ in such a way that for all $\square_l \psi \in \text{REQ}'(d_{r+1})$, $\psi \in A'$ and for all $\psi \in A'$, if $\psi = \diamond_r \xi$ or $\psi = \square_r \xi$, then $\psi \in \text{REQ}'(d_{r+1})$. Finally, add an immediate successor n' to the last node in B decorated with $\langle [d_j, d_{r+1}], A', \text{REQ}', \mathbb{D}' \rangle$.
2. *Left step rule:* if there exists an active node $n \in B$, with $\nu(n) = \langle [d_i, d_j], A_n, \text{REQ}_n, \mathbb{D}_n, x \rangle$ and a non-fulfilled past existential formula in A_n , then extend D_B to $D' = D_B \cup \{d_{l-1}\}$, with $d_{l-1} < d_l$. Then, take an atom A' such that $A' \text{ LR}_\varphi A_n$ and extend REQ_B to $\text{REQ}' : D' \mapsto \text{REQ}_\varphi$ in such a way that for all $\square_r \psi \in \text{REQ}'(d_{l-1})$, $\psi \in A'$ and for all $\psi \in A'$, if $\psi = \diamond_l \xi$ or $\psi = \square_l \xi$, then $\psi \in \text{REQ}'(d_{l-1})$. Finally, add an immediate successor n' to the last node in B decorated with $\langle [d_{l-1}, d_i], A', \text{REQ}', \mathbb{D}' \rangle$.
3. *Fill-in rule:* if there exist two points $d_i \leq d_j$ such that there are no nodes in B decorated with the interval $[d_i, d_j]$ and there exists a decoration $\langle [d_i, d_j], A', \text{REQ}_B, \mathbb{D}_B \rangle$, then expand B by adding an immediate successor n' , with such a decoration, to the last node in B .

All rules expand the branch B with a new node. However, while the left and right

step rules add a new point d to D_B and decorate the new node with a new interval beginning or ending at d , the fill-in rule decorates it with a new interval whose endpoints already belong to D_B .

Expansion strategy. Given a decorated tree \mathcal{T} and a branch B , let d_l and d_r be the least and the greatest point in \mathbb{D}_B , respectively. We say that B is *right-blocked* if $\text{REQ}_B(d_r)$ occurs $m + 1$ times in D_B , while it is *left-blocked* if $\text{REQ}_B(d_l)$ occurs $m + 1$ times in D_B . A branch is *blocked* if it is both left and right blocked.

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 decoration, add an immediate successor to the last node in B ;
2. if the fill-in rule is not applicable and there exist two points $d_i \leq d_j \in D_B$ such that there are no nodes in B decorated with $[d_i, d_j]$, *close* the branch;
3. if B is not right-blocked and the right-step rule is applicable, then apply it to B and, for every possible choice for the decoration, add an immediate successor to the last node in B ;
4. if B is not left-blocked and the left-step rule is applicable, then apply it to B and, for every possible choice for the decoration, add an immediate successor to the last node in B .

Tableau. Let φ be the formula to be checked for satisfiability and let $\langle [d_0, d_0], A_1, \text{REQ}_1, \{d_0\} \rangle, \dots, \langle [d_0, d_0], A_k, \text{REQ}_k, \{d_0\} \rangle$ be the set of decorations with $\diamond_r \varphi \in \text{REQ}_i(d_0)$. The *initial tableau* for φ consists of the root, with the empty decoration, and k immediate successors n_1, \dots, n_k . For each $1 \leq i \leq k$, n_i is decorated by $\langle [d_0, d_0], A_i, \text{REQ}_i, \{d_0\} \rangle$. 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 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 does not contain active nodes (finite model case);
2. B is right blocked and there exists at least one formula $\diamond_r \psi$ not fulfilled in B (right unbounded model case). Moreover, let d_r be the greatest point in D_B . By the blocking condition, $\text{REQ}_B(d_r)$ is repeated $m + 1$ times in D_B . Let d_k be the greatest point in D_B , with $d_k < d_r$, such that $\text{REQ}_B(d_k) = \text{REQ}_B(d_r)$. The set $\{d_{k+1}, \dots, d_r\}$, called *fulfilling right period*, satisfies the following conditions:
 - (a) for all $d_i, d_j \in \{d_{k+1}, \dots, d_r\}$, there exists an atom A_{ij} such that (i) $\pi \notin A_{ij}$, (ii) for all $\square_r \psi \in \text{REQ}_B(d_i)$, $\psi \in A_{ij}$, and (iii) for all $\square_l \psi \in \text{REQ}_B(d_j)$, $\psi \in A_{ij}$;

- (b) for all $d_i \in \{d_{k+1}, \dots, d_r\}$ and $\diamond_r \psi \in \text{REQ}_B(d_i)$ not fulfilled in B , there exist a point $d_j \in \{d_{k+1}, \dots, d_r\}$ and an atom A_{ij} such that (i) $\pi \notin A_{ij}$, (ii) $\psi \in A_{ij}$, (iii) for all $\square_r \xi \in \text{REQ}_B(d_i)$, $\xi \in A_{ij}$, and (iv) for all $\square_l \xi \in \text{REQ}_B(d_j)$, $\xi \in A_{ij}$;
 - (c) for all $d_i \leq d_k$ such that $\text{REQ}_B(d_i)$ does not occur in the right period, all \diamond_r -formulae in $\text{REQ}_B(d_i)$ are fulfilled in B .
3. B is left blocked and there exists at least one formula $\diamond_l \psi$ not fulfilled in B (left unbounded model case). Moreover, let d_l be the smallest point in D_B . By the blocking condition, $\text{REQ}_B(d_l)$ is repeated $m+1$ times in D_B . Let d_k be the smallest point in D_B , with $d_k > d_l$, such that $\text{REQ}_B(d_k) = \text{REQ}_B(d_l)$. The set $\{d_l, \dots, d_{k-1}\}$, called *fulfilling left period*, satisfies the following conditions:
- (a) for all $d_i, d_j \in \{d_l, \dots, d_{k-1}\}$, there exists an atom A_{ij} such that (i) $\pi \notin A_{ij}$, (ii) for all $\square_r \psi \in \text{REQ}_B(d_i)$, $\psi \in A_{ij}$, and (iii) for all $\square_l \psi \in \text{REQ}_B(d_j)$, $\psi \in A_{ij}$;
 - (b) for all $d_i \in \{d_l, \dots, d_{k-1}\}$ and $\diamond_l \psi \in \text{REQ}_B(d_i)$ not fulfilled in B , there exists a point $d_j \in \{d_l, \dots, d_{k-1}\}$ and an atom A_{ji} such that (i) $\pi \in A_{ji}$, (ii) $\psi \in A_{ji}$, (iii) for all $\square_r \xi \in \text{REQ}_B(d_j)$, $\xi \in A_{ji}$, and (iv) for all $\square_l \xi \in \text{REQ}_B(d_i)$, $\xi \in A_{ji}$;
 - (c) for all $d_i \geq d_k$ such that $\text{REQ}_B(d_i)$ does not occur in the left period, all \diamond_l -formulae in $\text{REQ}_B(d_i)$ are fulfilled in B .
4. if B is both right and left blocked, Conditions 2. and 3. must hold.

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} .

5.2.1 Soundness and completeness

Soundness and completeness of the proposed method can be proved as follows. Soundness is proved by showing how to construct a fulfilling LIS satisfying φ from a fulfilling branch B in a tableau \mathcal{T} for φ (by Theorem 5.7, it follows that φ has a model). The proof must encompass both the case of non-blocked branches (finite case) and of blocked ones (infinite case). Proving completeness consists in showing that for any satisfiable formula φ , there exists a fulfilling branch B in any tableau \mathcal{T} for φ . Given a model for φ and the corresponding fulfilling LIS \mathbf{L} , we prove the existence of a fulfilling branch in \mathcal{T} by exploiting Theorems 5.11 and 5.14.

Theorem 5.15 (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 . We first consider the LIS $\mathbf{L}_B = \langle \mathbb{D}_B, \mathbb{I}(\mathbb{D}_B)^+, \mathcal{L}_B \rangle$, where \mathcal{L}_B is such that, for every $[d_i, d_j] \in \mathbb{I}(\mathbb{D}_B)^+$, $\mathcal{L}_B([d_i, d_j]) = A_n$, with n being the unique node in B decorated with $\langle [d_i, d_j], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$. Given

the expansion rules of the tableau, we have that \mathbf{L}_B is a LIS, but it is not necessarily fulfilling. Four cases may arise.

B does not contains active nodes (*finite model case*). In this case, all \diamond_r and \diamond_l -formulae that occur in B are fulfilled in B and thus in \mathbf{L}_B . By the definition of initial tableau we have that $\diamond_r\varphi \in \text{REQ}_B(d_0)$. Hence, φ is satisfied in \mathbf{L}_B .

B is right blocked and it contains at least one non-fulfilled \diamond_r formula, while all \diamond_l -formulae are fulfilled in B (*right-unbounded model case*). In this case, we extend \mathbf{L}_B to a right unbounded LIS \mathbf{L}' where all \diamond_r -formulae are fulfilled. Let d_r be the greatest point of D_B and d_k be the greatest point in D_B such that $d_k < d_r$ and $\text{REQ}_B(d_k) = \text{REQ}_B(d_r)$. We extend D_B to D' by putting an infinite sequence of point d_{r+1}, d_{r+2}, \dots after d_r and we build the right-periodic LIS $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}')^+, \mathcal{L}' \rangle$ as follows:

- for all intervals $[d, d'] \in \mathbb{I}(\mathbb{D}_B)^+$, $\mathcal{L}'([d, d']) = \mathcal{L}_B([d, d'])$;
- for all points $d_{r+h} > d_r$, let $d_q = d_{k+(h \bmod (r-k))}$. We put $\text{REQ}^{\mathbf{L}'}(d_{r+h}) = \text{REQ}^{\mathbf{L}_B}(d_q)$ and $\mathcal{L}'([d_{r+h}, d_{r+h}]) = \mathcal{L}_B([d_q, d_q])$;
- for every point $d_{r+h} > d_r$, we fulfill the \diamond_l -formulae in $\text{REQ}^{\mathbf{L}'}(d_{r+h})$ as follows. First, for every $d_i < d_k$ such that $\text{REQ}^{\mathbf{L}'}(d_i)$ does not occur in the period, we put $\mathcal{L}'([d_i, d_{r+h}]) = \mathcal{L}_B([d_i, d_{k+(h \bmod (r-k))}])$. Then, for every formula $\diamond_l\psi \in \text{REQ}^{\mathbf{L}'}(d_{r+h})$ which has not been fulfilled yet, we consider the point $d_{k+(h \bmod (r-k))}$. Since in B all \diamond_l -formulae are fulfilled, there exists an interval $[d, d_{k+(h \bmod (r-k))}]$ such that $\psi \in \mathcal{L}_B([d, d_{k+(h \bmod (r-k))}])$. Two cases may arise. Either $\text{REQ}^{\mathbf{L}'}(d)$ occurs in the period or it does not. If $\text{REQ}^{\mathbf{L}'}(d)$ does not occur in the period, then $\mathcal{L}'([d, d_{r+h}]) = \mathcal{L}_B([d, d_{k+(h \bmod (r-k))}])$ and $\diamond_l\psi \in \text{REQ}^{\mathbf{L}'}(d_{r+h})$ is already fulfilled. If $\text{REQ}^{\mathbf{L}'}(d)$ occurs in the period, we take the greatest point $d' < d_{r+h}$ such that $\text{REQ}^{\mathbf{L}'}(d') = \text{REQ}^{\mathbf{L}'}(d)$ and the labeling of the interval $[d', d_{r+h}]$ has not been defined yet, and we put $\mathcal{L}'([d', d_{r+h}]) = \mathcal{L}_B([d, d_{k+(h \bmod (r-k))}])$. By making such a choice for d' , we guarantee that there always exist infinitely many points $d'' > d$ with the same set of requests of d_{r+h} such that the labeling of $[d, d'']$ is still undefined;
- for every point $d \in D'$ and every $\diamond_r\psi \in \text{REQ}^{\mathbf{L}'}(d)$ which has not been fulfilled yet, proceed as follows. By Condition 2.(c) of the definition of fulfilling branch, there exists a point $d_{k+1} \leq d_i \leq d_r$ such that $\text{REQ}_B(d) = \text{REQ}_B(d_i)$. Hence, by Condition 2.(b), there exist a point $d_j \in \{d_{k+1}, \dots, d_r\}$ and an atom A_{ij} such that $\pi \notin A_{ij}$, $\psi \in A_{ij}$, for all $\square_r\xi \in \text{REQ}_B(d_i)$, $\xi \in A_{ij}$, and for all $\square_l\xi \in \text{REQ}_B(d_j)$, $\xi \in A_{ij}$. By the definition of \mathbf{L}' , we have that there exist infinitely many points $d_n \geq d_r$ in D' such that $\text{REQ}^{\mathbf{L}'}(d_n) = \text{REQ}_B(d_j)$. We can take one of such points d_n such that $\mathcal{L}([d, d_n])$ has not been defined yet and put $\mathcal{L}([d, d_n]) = A_{ij}$;

- once we have fulfilled all diamond formulae in $\text{REQ}^{\mathbf{L}'}(d)$, for all $d \in D'$, we define the labeling of the remaining intervals $[d, d']$, where $d' > d_r$. Since B is fulfilling, we can always define $\mathcal{L}'([d, d'])$ by exploiting Condition 2.(b) for fulfilling branches.

B is left blocked and it contains at least one non-fulfilled \diamond_l formula, while all \diamond_r -formulae are fulfilled in B (*left-unbounded model case*). In this case we proceed as in the right-unbounded model case to extend \mathbf{L}_B to a left unbounded LIS where all \diamond_r -formulae are fulfilled.

B is left and right blocked and it contains both \diamond_r and \diamond_l non-fulfilled formulae (*unbounded model case*). We apply the construction for the right unbounded model case and that for the left unbounded model case to build an unbounded LIS where all diamond formulae are fulfilled. \square

Theorem 5.16 (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 $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{L} \rangle$ be a fulfilling LIS satisfying φ , whose existence is guaranteed by Theorem 5.7. Without loss of generality, we may assume that \mathbf{L} respects the constraints of Theorem 5.11, if it is finite, and of Theorem 5.14, if it is infinite. Furthermore, we assume that $\diamond_r \varphi \in \text{REQ}^{\mathbf{L}}(d_0)$. Given a linear order $\mathbb{D}' \subseteq \mathbb{D}$, we denote with $\text{REQ}^{\mathbf{L}}|_{\mathbb{D}'}$ the restriction of $\text{REQ}^{\mathbf{L}}$ to the intervals in $\mathbb{I}(\mathbb{D}')^+$. We prove that 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 $\langle [d_j, d_k], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, then $A_n = \mathcal{L}([d_j, d_k])$ and $\text{REQ}_n = \text{REQ}^{\mathbf{L}}|_{\mathbb{D}_n}$.* 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 $B_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle \cdot \langle [d_0, d_0], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, with $A_n = \mathcal{L}([d_0, d_0])$ and $\text{REQ}_n = \text{REQ}^{\mathbf{L}}|_{\{d_0\}}$.

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

- **The right-step rule has been applied.**

Let n be the active node, decorated with $\langle [d_j, d_l], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, to which the right-step rule has been applied to. By the inductive hypothesis, $A_n = \mathcal{L}([d_j, d_l])$ and $\text{REQ}_n = \text{REQ}^{\mathbf{L}}|_{\mathbb{D}_n}$. Let $D' = \{d_{-h}, \dots, d_{k+1}\}$. Since \mathbf{L} is a LIS, $\mathcal{L}([d_j, d_l]) \text{LR}_\varphi \mathcal{L}([d_l, d_{k+1}])$ and $\text{REQ}^{\mathbf{L}}|_{\mathbb{D}'}$ is a possible extension of REQ_n . Hence, there must exist in \mathcal{T} a successor n' of the last node of B_i decorated with $\langle [d_l, d_{k+1}], \mathcal{L}([d_l, d_{k+1}]), \text{REQ}^{\mathbf{L}}|_{\mathbb{D}'}, \mathbb{D}' \rangle$. Let $B_{i+1} = B_i \cdot n'$. Since the step rule can be applied only to non-closed branches (and it does not close any branch), B_{i+1} is non-closed.

- **The left-step rule has been applied.**

Let n be the active node, decorated with $\langle [d_j, d_i], A_n, \text{REQ}_n, \mathbb{D}_n \rangle$, to which the left-step rule has been applied to. By proceeding as in the case of the right-step rule, we can extend B_i to a non closed branch B_{i+1} that respects the property.

- **The fill-in rule has been applied.**

Let $d_j \leq d_i$ be the points in D_{B_i} such that there are no nodes in B_i decorated with $[d_j, d_i]$. By the inductive hypothesis (and by the definition of LIS), we have that $\langle [d_j, d_i], \mathcal{L}([d_j, d_i]), \text{REQ}_{B_i}, \mathbb{D}_{b_i} \rangle$ is a possible decoration. Hence, there must exist in \mathcal{T} a successor n' of the last node of B_i decorated with $\langle [d_i, d_j], \mathcal{L}([d_j, d_i]), \text{REQ}_{B_i}, \mathbb{D}_{b_i} \rangle$. Let $B_{i+1} = B_i \cdot n'$. 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, one of the following 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 $\diamond_r \psi \in A_n$ (resp. $\diamond_l \psi \in A_n$), there exists a node n' fulfilling it. Hence, B is a fulfilling branch.
- *B is right-blocked.* This implies that $\text{REQ}_B(d_k)$ is repeated $m + 1$ times in B . Since B is decorated coherently to \mathbf{L} , by Theorem 5.11, we can assume \mathbf{L} to be infinite. Let $d_j < d_k$ be the greatest point in D_B such that $\text{REQ}_B(d_j) = \text{REQ}_B(d_k)$. We have that \mathbf{L} is ultimately periodic, with right prefix $r = k - j$, since (by Theorem 5.14) the only set of requests which has been repeated $m + 1$ times in B is the one associated with the first point in the right period. Furthermore, we have that there are exactly $fp + f$ repetitions of the right period in B . This allows us to exploit the structural properties of \mathbf{L} to prove that B is fulfilling.

For every pair of points $d, d' \in \{d_j, \dots, d_k\}$, we have that $d, d' \in \text{Inf}(\mathbf{L})$. Hence, there exist infinitely many points d'' in \mathbf{L} such that $\text{REQ}_B(d'') = \text{REQ}_B(d')$ and $d < d''$. Let d'' be one of such points. We can choose the atom $A'' = \mathcal{L}([d, d''])$ to satisfy Condition 2.(a) of the definition of fulfilling branch.

For every point $d \in \{d_j, \dots, d_k\}$ and for every formula $\diamond_r \psi \in \text{REQ}_B(d)$, 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 $\diamond_r \psi$ is fulfilled in B . Otherwise, there exists a point d_m , with $d_i \leq d_m \leq d_k$, such that $\text{REQ}_B(d'') = \text{REQ}_B(d_m)$. Hence, the atom $A' = \mathcal{L}([d', d''])$ can be chosen in order to satisfy Condition 2.(b) of the definition of fulfilling branch.

For every point $d \in D$ such that $\text{REQ}_B(d_i)$ does not occur in the period, we have that $d \in \text{Fin}(\mathbf{L})$. Hence, by Theorem 5.14, we have that every formula $\diamond_r \psi \in \text{REQ}_B(d)$ is fulfilled by an interval $[d, d']$ such that $d' \leq d_{i+(fp+f)r}$. Since d_k corresponds to the first point of the $(fp + f + 1)$ -th occurrence of the right period, we have that $d' < d_k$ and hence $\diamond_r \psi$ is fulfilled in B . This shows that B respects Condition 2.(c) of the definition of fulfilling branch.

- The cases when B is left-blocked, or both right and left-blocked can be proved as the case when B is a right-blocked branch. \square

5.2.2 Computational complexity

In this section we provide a precise characterization of the computational complexity of the satisfiability problem for $\text{PNL}^{\pi+}$.

As for the computational complexity of the proposed decision procedure, observe that, by the blocking condition, after at most $|\text{REQ}_\varphi| \cdot m + 1$ applications of the step rules, the expansion strategy cannot be applied anymore to a branch. Moreover, given a branch B , between two successive applications of the step rules, the fill-in rule can be applied at most k times, being k the number of points in D_B (as a matter of fact, k is exactly the number of applications of the step rules up to that point). Since $m = 2fp + p \leq 2 \cdot |\text{TF}(\varphi)|^2 + |\text{TF}(\varphi)|$, we have that m is polynomial in the length of φ , while $|\text{REQ}_\varphi|$ is exponential in it. If $|\varphi| = n$, the length of any branch B of a tableau \mathcal{T} for φ is bounded by $(|\text{REQ}_\varphi| \cdot (2 \cdot |\text{TF}(\varphi)|^2 + |\text{TF}(\varphi)|))^2 = 2^{O(n)}$, that is, the length of a branch is exponential in $|\varphi|$. This implies that the satisfiability problem for $\text{PNL}^{\pi+}$ can be solved by a nondeterministic algorithm that guesses a fulfilling branch B for the formula φ in nondeterministic exponential time.

To give a NEXPTIME lower bound to the complexity of the satisfiability problem for $\text{PNL}^{\pi+}$ we can exploit the computational complexity results for RP NL given in the previous chapter. NEXPTIME-hardness of RP NL is proved by reducing the exponential tiling problem to the satisfiability problem for RP NL. Since RP NL is a fragment of $\text{PNL}^{\pi+}$, the reduction presented in Chapter 3 proves NEXPTIME-hardness of $\text{PNL}^{\pi+}$ as well.

Theorem 5.17. *The satisfiability problem for $\text{PNL}^{\pi+}$ is NEXPTIME-complete.*

5.3 A tableau for PNL^+ and PNL^-

As in the case of RP NL, the tableau-based decision procedure for $\text{PNL}^{\pi+}$ presented in the previous sections can be easily adapted to a decision procedure for PNL^+ by simply ignoring the conditions on the π operator. To adapt it to the case of PNL^- , we need to define the notion of *strict* φ -labelled interval structure.

Definition 5.18. A *strict φ -labelled interval structure* (strict LIS, for short) is a pair $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-, \mathcal{L} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle$ is a strict interval structure and $\mathcal{L} : \mathbb{I}(\mathbb{D})^- \rightarrow A_\varphi$ is a *labelling function* such that, for every pair of neighboring intervals $[d_i, d_j]$, $[d_j, d_k] \in \mathbb{I}(\mathbb{D})^-$, $\mathcal{L}([d_i, d_j]) LR_\varphi \mathcal{L}([d_j, d_k])$.

Theorems 5.7, 5.11, and 5.14 hold also for strict LISs and thus we can easily tailor the tableau-based decision method for $\text{PNL}^{\pi+}$ to PNL^- by ignoring the constraints on the π operator and by rewriting the fill-in rule as follows.

3. *Fill-in rule:* if there exist two points $d_i < d_j$ such that there are no nodes in B decorated with the interval $[d_i, d_j]$ and there exists a decoration $\langle [d_i, d_j], A' \rangle$,

$\text{REQ}_B, \mathbb{D}_B\rangle$, then expand B by adding an immediate successor n' , with such a decoration, to the last node in B .

The definition of initial tableau has to be modified as follows. Let φ be the formula to be checked for satisfiability and let $\langle [d_0, d_1], A_1, \text{REQ}_1, \{d_0, d_1\} \rangle, \dots, \langle [d_0, d_1], A_k, \text{REQ}_k, \{d_0, d_1\} \rangle$ be the set of decorations with $\langle A \rangle \varphi \in \text{REQ}_i(d_0)$. The *initial tableau* for φ consists of the root, with the empty decoration, and k immediate successors n_1, \dots, n_k . For each $1 \leq i \leq k$, n_i is decorated by $\langle [d_0, d_1], A_i, \text{REQ}_i, \{d_0, d_1\} \rangle$.

By contrast, the expansion strategy, the blocking condition and the definition of fulfilling branch remain unchanged. Termination, soundness, and completeness of the resulting tableau method for PNL^- can be proved as in the case of $\text{PNL}^{\pi+}$.

Finally, to prove the optimality of the tableau for RPNL^- , we can exploit the reduction given in Section 3.3, provided that we replace \diamond_r by $\langle A \rangle$ and \square_r by $[A]$.

Theorem 5.19. *The complexity of the satisfiability problem for PNL^- , over the integers, is NEXPTIME-complete.*

6

Decidability and expressiveness of PNL

In the previous chapters we have discussed the decidability problem for various Propositional Neighborhood Logics interpreted over specific structures like the naturals, the integers, and infinite trees. In this chapter we focus our attention on expressiveness and decidability issues for Propositional Neighborhood Logics from a general point of view [BGMS07].

First, we compare the expressive power of $\text{PNL}^{\pi+}$, PNL^+ , and PNL^- , and we show that $\text{PNL}^{\pi+}$ is strictly more expressive than PNL^+ and PNL^- . Then, we prove that the satisfiability problem for $\text{PNL}^{\pi+}$ over the class of all linear orders, as well as over some natural subclasses of it, such as the class of all well-orders and the class of all finite linear orders, can be decided in NEXPTIME by reducing it to the satisfiability problem for the two-variable fragment of first-order logic over the same classes of structures [Ott01]. This result extends the result of Chapter 5, where we have proved the decidability of $\text{PNL}^{\pi+}$ over the integers. Next, we focus our attention on expressive completeness, in the spirit of Kamp's theorem [Kam79]. Kamp proved the functional completeness of the *Since* (S) and *Until* (U) temporal logic with respect to first-order definable connectives over Dedekind-complete linear orders. This result has been later re-proved and generalized in several ways (see [IK89, GHR94]). In particular, Stavi extended Kamp's result to the class of all linear orders by adding the binary operators S' and U' (see [GHR94] for details), while Etessami et al. proved the functional completeness of the *future* (F) and *past* (P) temporal logic (TL[F,P] for short) with respect to the monadic two-variable fragment of first-order logic $\text{MFO}^2[<]$ over \mathbb{N} [EVW02]. As for interval-based logics, Venema showed the functional completeness of CDT with respect to the three-variable (with at most two of them free) fragment of first-order logic $\text{FO}_{x,y}^3[<]$ over all linear orders. Here we prove the expressive completeness of $\text{PNL}^{\pi+}$ with respect to the full two-variable fragment of first-order logic over various classes of linear orders. We conclude the chapter with a comparison of $\text{PNL}^{\pi+}$ expressive power with that of other HS fragments.

6.1 The two-variable fragment of first-order logic

In this section we give some basic definitions about first-order logic that will be used in the rest of the chapter. Let us denote by FO^2 (resp., $\text{FO}^2[=]$) the fragment of first-order logic (resp., first-order logic with equality) whose language uses only two distinct (possibly reused) variables. We denote its formulas by α, β, \dots . For example, the formula $\forall x(P(x) \rightarrow \forall y\exists xQ(x, y))$ 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 logic $\text{FO}^2[<]$ over a purely relational vocabulary $\{=, <, P, Q, \dots\}$ 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. The formulas of $\text{FO}^2[<]$ 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 $\alpha \in \text{FO}^2[<]$, that is, $\alpha = \alpha(x, y)$.

Formulas of $\text{FO}^2[<]$ are interpreted over *relational models* of the form $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, where \mathbb{D} 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 [Sco62] by 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 [BGG97]. Later on, Mortimer extended this result by including equality in the language [Mor75]. More recently, Grädel, Kolaitis, and Vardi improved Mortimer's result by lowering the complexity bound [GKV97]. Finally, by building on techniques from [GKV97] and taking advantage of an in-depth analysis of the basic 1-types and 2-types in $\text{FO}^2[<]$ -models, Otto proved the decidability of $\text{FO}^2[<]$ over the class of all linear orderings as well as on some natural subclasses of it [Ott01].

Theorem 6.1 ([Ott01]). *The satisfiability problem for formulas in $\text{FO}^2[<]$ is decidable in NEXPTIME on each of the classes of structures where $<$ is interpreted as (i) any linear ordering, (ii) any well-ordering, (iii) any finite linear ordering, and (iv) the linear ordering on \mathbb{N} .*

6.2 Comparing the expressive power of logics

In the following we will compare the expressive power of $\text{PNL}^{\pi+}$ with that of PNL^+ and PNL^- as well as with that of other classical/temporal logics. There are several ways to compare the expressive power of different modal languages/logics, e.g., 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 has been done in [GMS03b]). Here we compare the considered logics with respect to truth at a given element of a model. We distinguish three different cases: the case in which we compare two interval logics over the same class of models, e.g., $\text{PNL}^{\pi+}$ and PNL^+ , the case in which we compare strict and non-strict interval logics, e.g., PNL^- and $\text{PNL}^{\pi+}$, and the case in which we compare an interval logic with a first-order logic, e.g., $\text{PNL}^{\pi+}$ and $\text{FO}^2[<]$.

Given two interval logics L and L' interpreted over the same class of models \mathcal{C} , we say that L' is *at least as expressive as* L (with respect to \mathcal{C}), and we denote it by $L \preceq_{\mathcal{C}} L'$ (\mathcal{C} is omitted whenever it is 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 , $\mathbf{M}, [a, b] \models \varphi$ iff $\mathbf{M}, [a, b] \models \tau(\varphi)$. 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 over *strict* interval models with an interval logic L^+ interpreted over *non-strict* interval models, 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 letter $p \in \mathcal{AP}$, $[a, b] \in V^-(p)$ 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^- , $\mathbf{M}^-, [a, b] \models \varphi$ iff $\mathbf{M}^+, [a, b] \models \tau(\varphi)$ 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^+ , $\mathbf{M}^+, [a, b] \models \varphi$ iff $\mathbf{M}^-, [a, b] \models \tau'(\varphi)$, 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 over relational models. In this case, the above criteria are no longer adequate, since we need to compare logics which are interpreted over different types of models (interval models and relational models). We deal with this complication, by following the approach outlined by Venema in [Ven91]. 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 [Ven91].

Definition 6.2. Given an interval model $\mathbf{M} = \langle \mathbb{I}(\mathbb{D}), V_{\mathbf{M}} \rangle$, the corresponding relational model $\eta(\mathbf{M})$ is a pair $\langle \mathbb{D}, V_{\eta(\mathbf{M})} \rangle$, where for all $p \in \mathcal{AP}$, $V_{\eta(\mathbf{M})}(p) = \{(a, b) \in D \times D : [a, b] \in V_{\mathbf{M}}(p)\}$.

As a matter of fact, the above relational models can be viewed as ‘point’ models for logics over \mathbb{D}^2 and the above transformation as a mapping of propositional letters of the interval logic, interpreted over $\mathbb{I}(\mathbb{D})$, into propositional letters of the target logic, interpreted over \mathbb{D}^2 [Ven90, SS03].

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 letters p^{\leq} and p^{\geq} of the interval logic with every binary relation P .

Definition 6.3. 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 , and we denote it by $L_I \preceq_R L_{FO}$, 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} , and we denote it by $L_{FO} \preceq_R L_I$, 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} , and we denote it by $L_I \equiv_R L_{FO}$, if $L_I \preceq_R L_{FO}$ and $L_{FO} \preceq_R L_I$. $L_I \prec_R L_{FO}$ and $L_{FO} \prec_R L_I$ are defined in the usual way.

6.3 PNL $^{\pi+}$, PNL $^+$, and PNL $^-$ expressiveness

In this section we compare the relative expressive power of PNL $^{\pi+}$, PNL $^+$, and PNL $^-$. The comparison of the expressive power of PNL $^{\pi+}$ and PNL $^+$ is based on an application of the bisimulation game for modal logics [GO07]. More precisely, we exploit a game-theoretic argument to show that there exist two models that can be distinguished by a 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 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 $\mathbf{M}_1^+ = \langle \mathbb{I}(\mathbb{D}_1)^+, V_1 \rangle$, which 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 $[a_1, b_1] \in \mathbb{I}(\mathbb{D}_1)^+$. The game is played by two players, Player I and Player II. If after any given round the current position is not a local isomorphism between the submodels of \mathbf{M}_0^+ and \mathbf{M}_1^+ induced by the corresponding configuration, Player I wins the game; otherwise, Player II wins.

At every round, given a current configuration $([a_0, b_0], [a_1, b_1])$, Player I plays one of the following two moves:

\diamond_r -**move**: Player I chooses \mathbf{M}_i^+ , with $i \in \{0, 1\}$, and an interval $[b_i, c_i]$;

\diamond_l -**move**: Player I chooses \mathbf{M}_i^+ , with $i \in \{0, 1\}$, and an interval $[c_i, a_i]$.

In the first case, Player II replies by choosing an interval $[b_{1-i}, c_{1-i}]$, which leads to the new configuration $([b_0, c_0], [b_1, c_1])$; in the other case, Player II chooses an interval $[c_{1-i}, a_{1-i}]$, which leads to the new configuration $([c_0, a_0], [c_1, a_1])$. Roughly speaking, 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 [GO07]. The following key property of the k -round PNL⁺ bisimulation game directly follows from standard results for bisimulation games in modal logics.

Proposition 6.4. *Let \mathcal{P} be a finite set of propositional letters. 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 $([a_0, b_0], [a_1, b_1])$, if and only if $[a_0, b_0]$ and $[a_1, b_1]$ satisfy the same formulas of PNL⁺ over \mathcal{P} with operator depth at most k .*

We exploit Proposition 6.4 to prove that the π operator of PNL^{π+} cannot be expressed in PNL⁺. We choose two models \mathbf{M}_0^+ and \mathbf{M}_1^+ that can be distinguished with a PNL^{π+} 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 6.5. *The interval operator 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 $[1, 1]$ can be easily distinguished in PNL^{π+}, 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 $([0, 1], [1, 1])$, as follows: if Player I plays a \diamond_r -move on a given structure, then Player II arbitrarily chooses 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 arbitrarily chooses 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 a local isomorphism. \square

The next theorem shows that PNL⁻ is strictly less expressive than PNL^{π+} as well.

Theorem 6.6. $\text{PNL}^- \prec_I \text{PNL}^{\pi+}$.

Proof. We prove the claim by showing that $\text{PNL}^- \preceq_I \text{PNL}^{\pi+}$ and $\text{PNL}^{\pi+} \not\preceq_I \text{PNL}^-$. To prove the former, we provide a translation τ from PNL^- to $\text{PNL}^{\pi+}$. Consider the mapping τ_0 defined as follows:

$$\begin{aligned} \tau_0(p) &= p & \tau_0(\langle A \rangle \varphi) &= \diamond_r(\neg\pi \wedge \tau_0(\varphi)) \\ \tau_0(\neg\varphi) &= \neg\tau_0(\varphi) & \tau_0(\langle \bar{A} \rangle \varphi) &= \diamond_l(\neg\pi \wedge \tau_0(\varphi)) \\ \tau_0(\varphi_1 \vee \varphi_2) &= \tau_0(\varphi_1) \vee \tau_0(\varphi_2) \end{aligned}$$

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 that for any interval $[a, b]$ in \mathbf{M}^- and any PNL^- -formula φ , $\mathbf{M}^-, [a, b] \models \varphi$ if and only if $\mathbf{M}^+, [a, b] \models \tau(\varphi)$. The proof is an easy induction on the structure of φ . This proves that $\text{PNL}^- \preceq_I \text{PNL}^{\pi+}$.

To prove that $\text{PNL}^{\pi+} \not\preceq_I \text{PNL}^-$, suppose by contradiction that there exists a translation τ' from $\text{PNL}^{\pi+}$ to PNL^- such that, for any non-strict model \mathbf{M}^+ , any strict interval $[a, b]$, and any formula φ of $\text{PNL}^{\pi+}$, $\mathbf{M}^+, [a, b] \models \varphi$ iff $\mathbf{M}^-, [a, b] \models \tau'(\varphi)$, 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 that $\mathbf{M}_0^+, [0, 1] \models \Box_r p$, while $\mathbf{M}_1^+, [0, 1] \not\models \Box_r p$. 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 may conclude that $\mathbf{M}^-, [0, 1] \models \tau'(\Box_r p)$ and $\mathbf{M}^-, [0, 1] \not\models \tau'(\Box_r p)$, which is a contradiction. \square

Finally, we show that neither $\text{PNL}^+ \preceq_I \text{PNL}^-$ nor $\text{PNL}^- \preceq_I \text{PNL}^+$.

Theorem 6.7. *The expressive powers of PNL^+ and PNL^- are incomparable, that is, $\text{PNL}^- \not\preceq_I \text{PNL}^+$ and $\text{PNL}^+ \not\preceq_I \text{PNL}^-$.*

Proof. We first prove that $\text{PNL}^- \not\preceq_I \text{PNL}^+$. 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 the restriction of V_0 to $\mathbb{I}(\mathbb{Z} \setminus \{2\})^+$, 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 $[a, 2]$ (resp., $[2, b]$) in \mathbf{M}_0^+ , then Player II chooses the interval $[a, 1]$ (resp., $[1, b]$) in \mathbf{M}_1^+ . By contrast, 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 $\mathbf{M}_0^-, [0, 1] \models \langle A \rangle p$, while $\mathbf{M}_1^-, [0, 1] \not\models \langle A \rangle p$. Since \mathbf{M}_0^+ and \mathbf{M}_1^+ satisfy the same formulas over the interval $[0, 1]$, there cannot exist a translation τ' from PNL^- to PNL^+ such that $\mathbf{M}_0^+, [0, 1] \models \tau'(\langle A \rangle p)$ and $\mathbf{M}_1^+, [0, 1] \not\models \tau'(\langle A \rangle p)$.

As for $\text{PNL}^+ \not\preceq_I \text{PNL}^-$, we can exploit the very same proof we give to show that $\text{PNL}^{\pi+} \not\preceq_I \text{PNL}^-$ (it suffices to notice that $\Box_r p$ is a PNL^+ formula). \square

6.4 Decidability of PNL

In this section we prove the decidability of $\text{PNL}^{\pi+}$, and thus that of its proper fragments PNL^+ and PNL^- , by embedding it into the two-variable fragment of first-order logic interpreted over linearly ordered domains.

$\text{PNL}^{\pi+}$ can be translated into $\text{FO}^2[<]$ as follows. Let \mathcal{AP} be the set of propositional letters in $\text{PNL}^{\pi+}$. The signature for $\text{FO}^2[<]$ includes a binary relational symbol P for every propositional variable $p \in \mathcal{AP}$. The translation function $ST_{x,y}$, where x, y are two first-order variables, is defined as follows:

$$ST_{x,y}(\varphi) = x \leq y \wedge ST'_{x,y}(\varphi),$$

where

$$\begin{array}{ll} ST'_{x,y}(p) &= P(x, y) & ST'_{x,y}(\varphi \vee \psi) &= ST'_{x,y}(\varphi) \vee ST'_{x,y}(\psi) \\ ST'_{x,y}(\pi) &= (x = y) & ST'_{x,y}(\diamond_r \varphi) &= \exists x(y \leq x \wedge ST'_{y,x}(\varphi)) \\ ST'_{x,y}(\neg \varphi) &= \neg ST'_{x,y}(\varphi) & ST'_{x,y}(\diamond_l \varphi) &= \exists y(y \leq x \wedge ST'_{y,x}(\varphi)) \end{array}$$

Two variables are thus sufficient to translate $\text{PNL}^{\pi+}$ into $\text{FO}^2[<]$. As we will show later, this is not the case with other interval temporal logics, such as, for instance, HS and CDT. The next theorem proves that that $\text{FO}^2[<]$ is at least as expressive as $\text{PNL}^{\pi+}$ (η is the model transformation defined in Section 6.2).

Theorem 6.8. *For any $\text{PNL}^{\pi+}$ -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, as well as the cases of 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 such that $c \geq b$ and $\mathbf{M}^+, [b, c] \Vdash \psi$. 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 $a \leq b$ ($[a, b]$ in \mathbf{M}^+), we can conclude that $\eta(\mathbf{M}^+) \models ST_{x,y}(\diamond_r \psi)[x := a, y := b]$. The converse direction can be proved in a similar way. The case $\varphi = \diamond_l \psi$ is completely analogous and thus omitted. \square

Corollary 6.9. *A $\text{PNL}^{\pi+}$ -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 $\text{FO}^2[<]$ -models expanding linear orderings from \mathcal{C} .*

Since the above translation is polynomial in the size of the input formula, decidability of $\text{PNL}^{\pi+}$ follows from Theorem 6.1.

Corollary 6.10. *The satisfiability problem for $\text{PNL}^{\pi+}$ is decidable in NEXPTIME for each of the classes of non-strict interval structures built over (i) the class of all linear orderings, (ii) the class of all well-orderings, (iii) the class of all finite linear orderings, and (iv) the linear ordering on \mathbb{N} .*

This result can be extended to decide the satisfiability problem for $\text{PNL}^{\pi+}$ over any class of linear orderings, definable in $\text{FO}^2[<]$ within any of the above, e.g., the class of all (un)bounded (above, below) linear orderings or all (un)bounded above well-orderings, etc. By contrast, the decidability of the satisfiability problem for $\text{PNL}^{\pi+}$ on any of the classes of all discrete, dense, or Dedekind complete linear orderings is still open.

Since it holds that $\text{PNL}^+ \prec \text{PNL}^{\pi+}$ and $\text{PNL}^- \prec_I \text{PNL}^{\pi+}$, both PNL^+ and PNL^- are decidable in NEXPTIME (at least) over the same classes of orderings as $\text{PNL}^{\pi+}$. Moreover, a translation from PNL^+ to $\text{FO}^2[<]$ can be obtained from that for $\text{PNL}^{\pi+}$ by simply removing the rule for π , while a translation from PNL^- to $\text{FO}^2[<]$ can be obtained from that for $\text{PNL}^{\pi+}$ by removing the rule for π , by substituting $<$ for \leq , and by replacing \diamond_r (resp., \diamond_l) with $\langle A \rangle$ (resp., $\langle \bar{A} \rangle$).

The NEXPTIME-hardness of the satisfiability problem for $\text{PNL}^{\pi+}$, PNL^+ , and PNL^- can be proved by exploiting the very same reduction from the exponential tiling problem given for RPNL future fragments in Chapter 3.

Theorem 6.11. *The satisfiability problem for PNL^- , PNL^+ , and $\text{PNL}^{\pi+}$ interpreted in the class of all linear orderings, the class of all well-orderings, the class of all finite linear orderings, and the linear ordering on \mathbb{N} is NEXPTIME-complete.*

6.5 Expressive Completeness

In this section, we show that $\text{PNL}^{\pi+}$ is at least as expressive as $\text{FO}^2[<]$, that is, we show that every formula of $\text{FO}^2[<]$ can be translated into an equivalent formula of $\text{PNL}^{\pi+}$ (see Section 6.2). This allows us to conclude that $\text{PNL}^{\pi+}$ is as expressive as $\text{FO}^2[<]$. A similar result for CDT was given by Venema in [Ven91], where the expressive completeness of CDT with respect to $\text{FO}_{x,y}^3[<]$ (the fragment of first-order logic interpreted over linear orderings whose language features only three, possibly reused variables and at most two of them, x and y , can be free) was proved. Both results can be viewed as interval-based counterparts of Kamp's theorem for propositional point-based linear time temporal logic [Kam79].

The translation τ from $\text{FO}^2[<]$ to $\text{PNL}^{\pi+}$ is given by the following table:

Basic formulas	Non-basic formulas
$\tau[x, y](x = x) = \tau[x, y](y = y) = \top$	$\tau[x, y](\neg\alpha) = \neg\tau[x, y](\alpha)$
$\tau[x, y](x = y) = \tau[x, y](y = x) = \pi$	$\tau[x, y](\alpha \vee \beta) = \tau[x, y](\alpha) \vee \tau[x, y](\beta)$
$\tau[x, y](y < x) = \perp$	$\tau[x, y](\exists x\beta) =$
$\tau[x, y](x < y) = \neg\pi$	$\quad \diamond_r(\tau[y, x](\beta)) \vee \square_r \diamond_l(\tau[x, y](\beta))$
$\tau[x, y](P(x, x)) = \diamond_l(\pi \wedge p^{\leq} \wedge p^{\geq})$	$\tau[x, y](\exists y\beta) =$
$\tau[x, y](P(y, y)) = \diamond_r(\pi \wedge p^{\leq} \wedge p^{\geq})$	$\quad \diamond_l(\tau[y, x](\beta)) \vee \square_l \diamond_r(\tau[x, y](\beta))$
$\tau[x, y](P(x, y)) = p^{\leq}$	
$\tau[x, y](P(y, x)) = p^{\geq}$	

As stated by Theorem 6.13 below, every $\text{FO}^2[<]$ -formula $\alpha(x, y)$ is mapped into two distinct $\text{PNL}^{\pi+}$ -formulas $\tau[x, y](\alpha)$ and $\tau[y, x](\alpha)$. The first one captures all and only the models of $\alpha(x, y)$ where $x \leq y$ (if any), while the second one captures all and only the models of $\alpha(x, y)$ where $y \leq x$ (if any).

Example 6.12. Consider the formula $\alpha = \exists x \neg \exists y (x < y)$, which constrains the model to be right-bounded. Let $\beta = \exists y (x < y)$. We have that

$$\begin{aligned} \tau[x, y](\beta) &= \diamond_l(\tau[y, x](x < y)) \vee \square_l \diamond_r(\tau[x, y](x < y)) = \\ &= \diamond_l \perp \vee \square_l \diamond_r \neg \pi \quad (\equiv \square_l \diamond_r \neg \pi) \end{aligned}$$

and that

$$\begin{aligned} \tau[y, x](\beta) &= \diamond_r(\tau[x, y](x < y)) \vee \square_r \diamond_l(\tau[y, x](x < y)) = \\ &= \diamond_r \neg \pi \vee \square_r \diamond_l \perp \quad (\equiv \diamond_r \neg \pi) \end{aligned}$$

The resulting translation of α is:

$$\begin{aligned} \tau[x, y](\alpha) &= \diamond_r(\tau[y, x](\neg \beta)) \vee \square_r \diamond_l(\tau[x, y](\neg \beta)) = \\ &= \diamond_r(\neg \tau[y, x](\beta)) \vee \square_r \diamond_l(\neg \tau[x, y](\beta)) = \\ &= \diamond_r \neg \diamond_r \neg \pi \vee \square_r \diamond_l \neg \square_l \diamond_r \neg \pi = \\ &= \diamond_r \square_r \pi \vee \square_r \diamond_l \diamond_l \square_r \pi \quad (\equiv \diamond_r \square_r \pi \vee \square_r \pi) \end{aligned}$$

which is a $\text{PNL}^{\pi+}$ -formula which constrains the model to be right-bounded.

Let $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$ be a $\text{FO}^2[<]$ -model and let $\zeta(\mathcal{A}) = \langle \mathbb{I}(\mathbb{D})^+, V_{\zeta(\mathcal{A})} \rangle$ be the corresponding $\text{PNL}^{\pi+}$ -model (see Section 6.2).

Theorem 6.13. *For every $\text{FO}^2[<]$ -formula $\alpha(x, y)$, every $\text{FO}^2[<]$ -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.
- $\alpha = (x < y)$. As for claim (i), $\mathcal{A} \models \alpha(a, b)$ iff $a < b$ iff $\zeta(\mathcal{A}), [a, b] \Vdash \neg \pi$. As for claim (ii) $\mathcal{A} \not\models \alpha(b, a)$, since $a \leq b$, and $\zeta(\mathcal{A}), [a, b] \not\Vdash \tau[y, x](x < y) (= \perp)$. 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 6.2).
- $\alpha = P(x, x)$. As for 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.

- $\alpha = \exists x\beta$. As for 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, $\zeta(\mathcal{A}), [a, b] \Vdash \square_r \diamond_l(\tau[x, y](\beta))$. Hence $\zeta(\mathcal{A}), [a, b] \Vdash \diamond_r(\tau[y, x](\beta)) \vee \square_r \diamond_l(\tau[x, y](\beta))$, 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 in a similar way.
- $\alpha = \exists y\beta$. Analogous to the previous case. \square

Corollary 6.14. *For every formula $\alpha(x, y)$ and every $\text{FO}^2[<]$ -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 6.15. We say that a $\text{PNL}^{\pi+}$ -model \mathbf{M} of the considered language is *synchronized on a pair of variables p^{\leq} and p^{\geq}* if these variables are equally true at any point interval $[a, a]$ in \mathbf{M} ; \mathbf{M} is *synchronized for a $\text{FO}^2[<]$ -formula α* if it is synchronized on every pair of variables p^{\leq} and p^{\geq} corresponding to a predicate p occurring in α ; \mathbf{M} is *synchronized* if it is synchronized on every pair p^{\leq} and p^{\geq} .

It is immediate that every model $\zeta(\mathcal{A})$, where \mathcal{A} is a $\text{FO}^2[<]$ -model, is synchronized. Conversely, every synchronized $\text{PNL}^{\pi+}$ -model \mathbf{M} can be represented as $\zeta(\mathcal{A})$ for some model \mathcal{A} for $\text{FO}^2[<]$: 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 6.13, 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 $\text{PNL}^{\pi+}$ -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 $\text{PNL}^{\pi+}$ as follows [GMS03b]:

$$[U]\varphi ::= \square_r \square_r \square_l \varphi \wedge \square_r \square_l \square_l \varphi \wedge \square_l \square_l \square_r \varphi \wedge \square_l \square_r \square_r \varphi.$$

Building on this observation, we associate with every $\text{FO}^2[<]$ -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 6.16. *For any FO $^2[<]$ -formula α , (i) α is valid in all FO $^2[<]$ -models iff $\sigma_v(\alpha)$ is a valid PNL $^{\pi+}$ -formula, and (ii) α is satisfiable in some FO $^2[<]$ -model iff $\sigma_s(\alpha)$ is a satisfiable PNL $^{\pi+}$ -formula.*

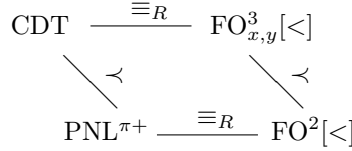


Figure 6.1: Expressive completeness results for interval logics.

In Figure 1 we put together the expressive completeness results for CDT and PNL $^{\pi+}$, using the notation introduced in Section 2. Since FO $^2[<]$ is a proper fragment of FO $^3_{x,y}[<]$, from the equivalences between CDT and FO $^3_{x,y}[<]$ and between PNL $^{\pi+}$ and FO $^2[<]$ it immediately follows that CDT is strictly more expressive than PNL $^{\pi+}$.

6.6 PNL $^{\pi+}$ and other HS fragments

In this section we explore the relationships between PNL $^{\pi+}$ and other fragments of HS. More precisely, we describe the fragments of HS which are fragments of PNL $^{\pi+}$ as well. To this end, we consider all other interval modalities of HS, namely, $\langle B \rangle$, $\langle E \rangle$, $\langle O \rangle$, $\langle D \rangle$, $\langle L \rangle$, and their transposes, which correspond to Allen's relations *begins*, *ends*, *overlaps*, *during*, and *after*, and their inverse relations. The semantics of such modalities can be given by their *standard translations* into first-order logic:

$$\begin{aligned}
 ST_{x,y}(\langle B \rangle \varphi) &= x \leq y \wedge \exists z (z < y \wedge ST_{x,z}(\varphi)) \\
 ST_{x,y}(\langle E \rangle \varphi) &= x \leq y \wedge \exists z (x < z \wedge ST_{z,y}(\varphi)) \\
 ST_{x,y}(\langle O \rangle \varphi) &= x \leq y \wedge \exists z (x < z < y \wedge \exists y (y < x \wedge ST_{y,z}(\varphi))) \\
 ST_{x,y}(\langle D \rangle \varphi) &= x \leq y \wedge \exists z (x < z < y \wedge \exists y (x < y \wedge ST_{y,z}(\varphi))) \\
 ST_{x,y}(\langle L \rangle \varphi) &= x \leq y \wedge \exists x (y < x \wedge \exists y ST_{x,y}(\varphi))
 \end{aligned}$$

The standard translation of $\langle L \rangle$ is a two-variable formula, while the standard translations of the other modalities are three-variable formulas. By taking advantage of the translation from FO $^2[<]$ to PNL $^{\pi+}$, $\langle L \rangle$ can be defined in PNL $^{\pi+}$ as follows:

$$\langle L \rangle \varphi = \diamond_r (\neg \pi \wedge \diamond_r \varphi).$$

We show that the other interval modalities cannot be defined in PNL $^{\pi+}$ by a game-theoretic argument similar to the one of Theorem 6.5. 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 of the k -round PNL $^+$ bisimulation game described in Section 6.3; the only difference

is that a configuration $([a_0, b_0], [a_1, b_1])$ constitutes a local isomorphism between \mathbf{M}_0^+ and \mathbf{M}_1^+ 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 proposition is analogous to Proposition 6.4.

Proposition 6.17. *Let \mathcal{P} be a finite set of propositional letters. For all $k \geq 0$, Player II has a winning strategy in the k -round $\text{PNL}^{\pi+}$ bisimulation game on \mathbf{M}_0^+ and \mathbf{M}_1^+ with initial configuration $([a_0, b_0], [a_1, b_1])$ if and only if $[a_0, b_0]$ and $[a_1, b_1]$ satisfy the same formulas of $\text{PNL}^{\pi+}$ over \mathcal{P} with operator depth at most k .*

We exploit Proposition 6.17 to prove that none of the interval modalities $\langle B \rangle$, $\langle E \rangle$, $\langle O \rangle$, and $\langle D \rangle$ is expressible in $\text{PNL}^{\pi+}$. 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 $\text{PNL}^{\pi+}$ bisimulation game.

Theorem 6.18. *Neither of $\langle B \rangle$, $\langle E \rangle$, $\langle O \rangle$, and $\langle D \rangle$ can be defined in $\text{PNL}^{\pi+}$.*

Proof. We will prove the claim for $\langle B \rangle$ and $\langle D \rangle$; the other cases are analogous. Consider the $\text{PNL}^{\pi+}$ -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 $\text{PNL}^{\pi+}$ 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, as follows: at any position, assuming that Player I has not won yet, if he chooses a \diamond_r -move then Player II arbitrarily chooses 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. \square

7

A relational approach to interval logics

In this chapter we present a relational proof system in the style of dual tableaux for relational logics associated with propositional interval logics and we prove that the systems enable us to verify satisfiability, validity, and entailment of these temporal logics. In constructing the systems we apply the method known for various non-classical logics, in particular for standard modal and temporal logics [Or195, Or196]. The key steps of the method are:

- Development of a relational logic RL_L appropriate for a given interval temporal logic L .
- Development of a validity preserving translation from the language of logic L into the language of logic RL_L .
- Construction of a proof system for RL_L such that for every formula φ of L , φ is valid in L iff its translation $\tau(\varphi)$ is provable in RL_L .

Each logic RL_L is based on the classical relational logic of binary relations, RL . RL_L is capable of expressing both binary relations holding between points of time and binary relations holding between time intervals. The proof systems developed in this chapter are extensions of the proof system for RL originated in [Or188] and further expanded in [GPO06a, Or196], that we have described in Chapter 2. In constructing deduction rules for our systems we follow the general principles of defining relational deduction rules presented in [MO02].

This chapter is a revised version of [BGPO06], and it is structured as follows. In Sections 7.1, 7.2, 7.3, and 7.4 we develop a relational proof system for the Halpern and Shoham's logic HS [HS91] in accordance with the three steps mentioned above. Next, in Section 7.5 we show how this system can be extended or modified in order to incorporate the remaining interval relations of Allen [All83, LM87] and/or other time orderings.

7.1 A relational logic for HS

The vocabulary of the language RL_{HS} consists of the pairwise disjoint sets listed below:

- a countable infinite set $\mathbb{IV} = \{i, j, k, \dots\}$ of *interval variables*;
- since intervals are meant to be certain pairs of points, to every interval variable i we associate two point variables denoted i_1, i_2 , with the intuition that $i = [i_1, i_2]$. We define the countable infinite set of *point variables* as $\mathbb{PV} = \{i_1, i_2 : i \in \mathbb{IV}\}$;
- a countable infinite set \mathbb{IRV} of *interval relational variables*;
- a set $\mathbb{PRC} = \{=, <\}$ of *point relational constants*;
- a set $\mathbb{IRC} = \{U, B, E\}$ of *interval relational constants*;
- a set $\mathbb{OP} = \{-, \cup, \cap, ;, ^{-1}\}$ of *relational operation symbols*.

The point constants $=$ and $<$ are intended to represent the identity relation and the ordering on the set of time points, respectively, while the interval constant U represent the universal relation between intervals. The unary operators $-$ and $^{-1}$ bind stronger than the binary \cup, \cap and $;$.

The specific relational operations of converse ($^{-1}$) and composition ($;$) are defined as usual. For binary relations R, S on a set D :

$$R^{-1} = \{(x, y) \in D \times D : (y, x) \in R\}$$

$$R; S = \{(x, y) \in D \times D : \exists z \in D [(x, z) \in R \wedge (z, y) \in S]\}$$

The syntax of RL_{HS} is defined as follows.

- The set of *point relational terms* \mathbb{PRT} is the smallest set of expressions that includes \mathbb{PRC} and is closed with respect to the operation symbols from \mathbb{OP} . In the following, we will use \neq instead of \neq and $\not<$ instead of $<$.
- The set of *interval relational terms* \mathbb{IRT} is the smallest set of expressions that includes $\mathbb{IRA} = \mathbb{IRV} \cup \mathbb{IRC}$ and is closed with respect to the operation symbols from \mathbb{OP} .
- The set of *point relational formulae* \mathbb{PRF} consists of expressions of the form $x R y$ where $x, y \in \mathbb{PV}$ and $R \in \mathbb{PRT}$.
- The set of *interval relational formulae* \mathbb{IRF} consists of expressions of the form $i R j$ where $i, j \in \mathbb{IV}$ and $R \in \mathbb{IRT}$.
- The set \mathbb{RF} of RL_{HS} -formulae (or, simply formulae if it is clear from the context), consists of expressions from $\mathbb{PRF} \cup \mathbb{IRF}$.
- R is said to be an *atomic relational term* whenever $R \in \mathbb{PRC} \cup \mathbb{IRA}$. $x R y$ is said to be an *atomic formula* whenever R is an atomic relational term.

Semantics

An RL_{HS} -model is a tuple $\mathcal{M} = (D, \mathbb{I}(D)^+, m)$, where D and $\mathbb{I}(D)^+$ are non-empty sets and $m: \text{PRT} \cup \text{IRT} \rightarrow 2^{D \times D} \cup 2^{\mathbb{I}(D)^+ \times \mathbb{I}(D)^+}$ is a meaning function which assigns binary relations on $D \times D$ to point relational terms and binary relations on $\mathbb{I}(D)^+ \times \mathbb{I}(D)^+$ to interval relational terms as follows:

- (1) $m(=)$ is the identity relation Id_D on D ;
- (2) $m(<)$ is a strict linear ordering on D , that is, for every $c, d, e \in D$ the following holds:

- (Irref) $(c, c) \notin m(<)$;
- (Trans) if $(c, d) \in m(<)$ and $(d, e) \in m(<)$, then $(c, e) \in m(<)$;
- (Lin) $(c, d) \in m(<)$ or $(d, c) \in m(<)$ or $(c, d) \in m(=)$;

- (3) m extends to all compound relational terms $R \in \text{PRT}$ as follows:

- $m(-R) = (D \times D) \setminus m(R)$;
- $m(R \cup S) = (m(R) \cup m(S))$;
- $m(R \cap S) = (m(R) \cap m(S))$;
- $m(R^{-1}) = m(R)^{-1}$;
- $m(R; S) = (m(R); m(S))$;

- (4) $\mathbb{I}(D)^+ = \{[c, d] \in D \times D : (c, d) \in m(< \cup =)\}$;

- (5) $m(U) = \mathbb{I}(D)^+ \times \mathbb{I}(D)^+$;

- (6) $m(B) = \{([c, d], [c', d']) \in m(U) : (c, c') \in m(=) \wedge (d', d) \in m(<)\}$;

- (7) $m(E) = \{([c, d], [c', d']) \in m(U) : (c, c') \in m(<) \wedge (d, d') \in m(=)\}$;

- (8) m extends to all compound relational terms $R \in \text{IRT}$ as in (3) except for the clause for $-R$: $m(-R) = m(U) \setminus m(R)$.

An RL_{HS} -valuation in a model $\mathcal{M} = (D, \mathbb{I}(D)^+, m)$ is any function $v: \text{PV} \cup \text{IV} \rightarrow D \cup \mathbb{I}(D)^+$ such that:

- if $x \in \text{PV}$ then $v(x) \in D$;
- if $i \in \text{IV}$ then $v(i) = [v(i_1), v(i_2)] \in \mathbb{I}(D)^+$.

We say that v satisfies a formula $x R y$ ($\mathcal{M}, v \models x R y$ for short) iff $(v(x), v(y)) \in m(R)$. A formula is *true* in \mathcal{M} whenever it is satisfied in \mathcal{M} by every valuation v . A formula is RL_{HS} -valid whenever it is true in every RL_{HS} -model.

7.2 Translation

In this section we present a translation of the formulae of HS into relational terms of RL_{HS} . We follow the general principle of translation of modal formulae presented in [Orl88]: modal formulae should be mapped into terms which represent right ideal relations, that is, relations satisfying the condition $R;U = R$. It is known that the Boolean operations preserve the property of being a right ideal relation, and the composition of any relation with a right ideal relation results in a right ideal relation. So our definition of translation enforces the property of having a right ideal translation for propositional variables. It follows that the property is guaranteed for the formulae built with the classical propositional connectives. Moreover, since the translation of modalities is defined as a composition of the constant denoting an accessibility relation with the translation of the formula to which the modality is applied, the translation results in a term representing a right ideal relation.

We consider the following translation function τ , that maps HS-formulae φ to RL_{HS} -formulae of the form $x R y$ as follows. Since we consider non-strict interval structures, we take as primitives only the interval modalities $\langle B \rangle$, $\langle E \rangle$, $\langle \bar{B} \rangle$, and $\langle \bar{E} \rangle$. As pointed out in Section 1.3.1, the modalities $\langle A \rangle$ and $\langle \bar{A} \rangle$ (that were originally included in HS) can be derived from the other ones in the non-strict semantics.

- for every propositional letter $p \in AP$, $\tau(p) = P;U$, where $P \in \mathbb{IRV}$ is a relational variable;
- $\tau(\neg\psi) = \neg\tau(\psi)$;
- $\tau(\psi_1 \vee \psi_2) = \tau(\psi_1) \cup \tau(\psi_2)$;
- $\tau(\langle B \rangle\psi) = B;\tau(\psi)$;
- $\tau(\langle E \rangle\psi) = E;\tau(\psi)$;
- $\tau(\langle \bar{B} \rangle\psi) = B^{-1};\tau(\psi)$;
- $\tau(\langle \bar{E} \rangle\psi) = E^{-1};\tau(\psi)$.

Lemma 7.1. *For every linear, non-strict HS-model \mathbf{M}^+ and for every HS-formula ψ there is an RL_{HS} -model \mathcal{M} such that ψ is true in \mathbf{M}^+ iff $i \tau(\psi) j$ is true in \mathcal{M} , with $i, j \in \mathbb{IV}$.*

Proof. Let ψ be an HS-formula, and let $\mathbf{M}^+ = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ be an HS-model. We define the corresponding RL_{HS} -model $\mathcal{M} = (D, \mathbb{I}(D)^+, m)$ as follows:

- $m(<) = \{(c, d) \in D : c < d\}$;
- $m(U) = \mathbb{I}(D)^+ \times \mathbb{I}(D)^+$ and $m(=) = \text{Id}_D$;
- for every $p \in AP$, $m(P) = \{([c, d], [c', d']) \in m(U) : [c, d] \in \mathcal{V}(p)\}$;
- $m(B) = \{([c, d], [c', d']) \in m(U) : c = c', d' < d\}$;

- $m(E) = \{([c, d], [c', d']) \in m(U) : c < c', d' = d\}$.

Given a valuation v we show by induction on the structure of ψ that the following property holds:

$$\mathbf{M}^+, v(i) \Vdash \psi \text{ iff } \mathcal{M}, v \models i \tau(\psi) j.$$

From that, we can conclude that ψ is true in \mathbf{M}^+ iff $i \tau(\psi) j$ is true in \mathcal{M} . By the definition of \mathcal{M} , such a property trivially holds for propositional letters. We prove it for formulae of the form $\psi_1 \vee \psi_2$, $\langle B \rangle \psi_1$ and $\langle \bar{E} \rangle \psi_1$. The other cases can be proved in a similar way.

- If $\psi = \psi_1 \vee \psi_2$ then $\mathbf{M}^+, v(i) \Vdash \psi_1 \vee \psi_2$ iff $\mathbf{M}^+, v(i) \Vdash \psi_1$ or $\mathbf{M}^+, v(i) \Vdash \psi_2$, iff, by inductive hypothesis, $\mathcal{M}, v \models i \tau(\psi_1) j$ or $\mathcal{M}, v \models i \tau(\psi_2) j$, iff $\mathcal{M}, v \models i (\tau(\psi_1) \cup \tau(\psi_2)) j$ iff $\mathcal{M}, v \models i \tau(\psi_1 \vee \psi_2) j$.
- If $\psi = \langle B \rangle \psi_1$ then $\mathbf{M}^+, v(i) \Vdash \langle B \rangle \psi_1$ iff there exists $c' < v(i_2)$ such that $\mathbf{M}^+, [v(i_1), c'] \Vdash \psi_1$, iff, by inductive hypothesis and by definition of \mathcal{M} , we have that $(v(i), [v(i_1), c']) \in m(B)$ and $([v(i_1), c'], [v(j_1), v(j_2)]) \in m(\tau(\psi_1))$, iff $\mathcal{M}, v \models i (B; \tau(\psi_1)) j$ iff $\mathcal{M}, v \models i \tau(\langle B \rangle \psi_1) j$.
- Finally, if $\psi = \langle \bar{E} \rangle \psi_1$ then $\mathbf{M}^+, v(i) \Vdash \langle \bar{E} \rangle \psi_1$ iff there exists $c' < v(i_1)$ such that $\mathbf{M}^+, [c', v(i_2)] \Vdash \psi_1$, iff, by inductive hypothesis and by definition of \mathcal{M} , $(v(i), [c', v(i_2)]) \in m(E^{-1})$ and $([c', v(i_2)], [v(j_1), v(j_2)]) \in m(\tau(\psi_1))$, iff $\mathcal{M}, v \models i (E^{-1}; \tau(\psi_1)) j$ iff $\mathcal{M}, v \models i \tau(\langle \bar{E} \rangle \psi_1) j$. \square

Lemma 7.2. *For every RL_{HS} -model \mathcal{M} and for every HS-formula ψ there is an HS-model \mathbf{M}^+ such that ψ is true in \mathbf{M}^+ iff $i \tau(\psi) j$ is true in \mathcal{M} , where $i, j \in \mathbb{IV}$.*

Proof. Let ψ be an HS-formula, and let $\mathcal{M} = (D, \mathbb{I}(D)^+, m)$ be an RL_{HS} -model. We define the corresponding HS-model $\mathbf{M}^+ = \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^+, \mathcal{V} \rangle$ as follows:

- $\mathbb{D} = \langle D, < \rangle$ is such that for all $c, d \in D$, $c < d$ iff $(c, d) \in m(<)$;
- for all $p \in \text{AP}$, $[c, d] \in \mathcal{V}(p)$ iff $([c, d], [c', d']) \in m(P; U)$, for all $[c', d'] \in \mathbb{I}(D)^+$.

Since $m(<)$ is a strict linear ordering on D , then $\langle D, < \rangle$ is a strict linear ordering, and thus \mathbf{M}^+ is correctly defined.

Given a valuation v we show by induction on the structure of ψ that the following property holds:

$$\mathcal{M}, v \models i \tau(\psi) j \text{ iff } \mathbf{M}^+, v(i) \Vdash \psi.$$

From that, we can conclude that $i \tau(\psi) j$ is true in \mathcal{M} iff ψ is true in \mathbf{M}^+ . By the definition of \mathbf{M}^+ , such a property trivially holds for propositional letters. We prove the required condition for the formulae of the form $\neg\psi_1$, $\langle E \rangle \psi_1$ and $\langle \bar{B} \rangle \psi_1$, the other cases are similar.

- If $\psi = \neg\psi_1$ then $\mathcal{M}, v \models i \tau(\neg\psi_1) j$ iff $\mathcal{M}, v \models i (-\tau(\psi_1)) j$ iff $\mathcal{M}, v \not\models i \tau(\psi_1) j$ iff, by inductive hypothesis, $\mathbf{M}^+, v(i) \not\Vdash \psi_1$ iff, $\mathbf{M}^+, v(i) \Vdash \neg\psi_1$.

- If $\psi = \langle E \rangle \psi_1$ then $\mathcal{M}, v \models i \tau(\langle E \rangle \psi_1) j$ iff $\mathcal{M}, v \models i (E; \tau(\psi_1)) j$ iff, there exists $[c', d'] \in \mathbb{I}(D)^+$ such that $(v(i), [c', d']) \in m(E)$ and $([c', d'], v(j)) \in m(\tau(\psi_1))$, iff, by the definition of $m(E)$ and by inductive hypothesis, $v(i_2) = d'$, $v(i_1) < c'$, and $\mathbf{M}^+, [c', d'] \Vdash \psi_1$, iff $\mathbf{M}^+, v(i) \Vdash \langle E \rangle \psi_1$.
- If $\psi = \langle \bar{B} \rangle \psi_1$ then $\mathcal{M}, v \models i \tau(\langle \bar{B} \rangle \psi_1) j$ iff $\mathcal{M}, v \models i (B^{-1}; \tau(\psi_1)) j$ iff, there exists $[c', d'] \in \mathbb{I}(D)^+$ such that $(v(i), [c', d']) \in m(B^{-1})$ and $([c', d'], v(j)) \in m(\tau(\psi_1))$, iff, by the definition of $m(B^{-1})$ and by inductive hypothesis, $v(i_1) = c'$, $d' > v(i_2)$, and $\mathbf{M}^+, [c', d'] \Vdash \psi_1$, iff $\mathbf{M}^+, v(i) \Vdash \langle \bar{B} \rangle \psi_1$. \square

From the above lemmas we obtain:

Theorem 7.3. *For every HS-formula ψ , ψ is HS-valid iff $i \tau(\psi) j$ is RL_{HS} -valid.*

7.3 The proof system for RL_{HS}

The proof system for the relational logic RL_{HS} presented in this section belongs to the family of dual tableau systems. It consists of axiomatic sets of formulae and rules which apply to finite sets of formulae. There are three groups of rules: rules which reflect definitions of the standard relational operations; rules which enable us to decompose interval relations into point relations according to the definitions recalled in Chapter 1; and rules which reflect the properties of the temporal ordering assumed in the models.

7.3.1 Decomposition rules

In the following, we say that a variable in a rule is *new* whenever it appears in a conclusion of the rule and does not appear in its premise.

Standard decomposition rules

The standard decomposition rules are the rules for the standard relational logic RL . Let $x, y, z \in \mathbb{P}\mathbb{V}$ and $R, S \in \mathbb{P}\mathbb{R}\mathbb{T}$ or $x, y, z \in \mathbb{I}\mathbb{V}$ and $R, S \in \mathbb{I}\mathbb{R}\mathbb{T}$.

$$\begin{array}{ll}
 (\cup) \quad \frac{x (R \cup S) y}{x R y, x S y} & (-\cup) \quad \frac{x -(R \cup S) y}{x -R y \mid x -S y} \\
 (\cap) \quad \frac{x (R \cap S) y}{x R y \mid x S y} & (-\cap) \quad \frac{x -(R \cap S) y}{x -R y, x -S y} \\
 (-) \quad \frac{x - -R y}{x R y} & \\
 (-1) \quad \frac{x R^{-1} y}{y R x} & (-^{-1}) \quad \frac{x -R^{-1} y}{y -R x}
 \end{array}$$

$$\begin{aligned}
(;) & \frac{x (R; S) y}{x R z, x (R; S) y \mid z S y, x (R; S) y} && z \text{ is any variable} \\
(-;) & \frac{x -(R; S) y}{x -R z, z -S y} && z \text{ is a new variable}
\end{aligned}$$

Decomposition rules from interval relations to point relations

For $i, j \in \mathbb{IV}$ and $R \in \mathbb{IRA}$:

$$\begin{aligned}
(R_1) & \frac{i R j}{i_1 = k_1, i R j \mid i_2 = k_2, i R j \mid k R j, i R j} && \text{with } k \text{ any interval variable.} \\
(R_2) & \frac{i R j}{j_1 = k_1, i R j \mid j_2 = k_2, i R j \mid i R k, i R j} && \text{with } k \text{ any interval variable.}
\end{aligned}$$

For $i, j \in \mathbb{IV}$:

$$\begin{aligned}
(B) & \frac{i B j}{i_1 = j_1, i B j \mid j_2 < i_2, i B j} && (-B) \frac{i -B j}{i_1 \neq j_1, j_2 \not< i_2, i -B j} \\
(E) & \frac{i E j}{i_2 = j_2, i E j \mid i_1 < j_1, i E j} && (-E) \frac{i -E j}{i_2 \neq j_2, i_1 \not< j_1, i -E j}
\end{aligned}$$

7.3.2 Specific rules

Rules for =

For $x, y \in \mathbb{PV}$ and $R \in \mathbb{PRC}$:

$$(=1) \frac{x R y}{x R z, x R y \mid y = z, x R y} \qquad (=2) \frac{x R y}{x = z, x R y \mid z R y, x R y}$$

with z any point variable.

Rules for <

For $x, y \in \mathbb{PV}$:

$$\begin{aligned}
(\text{Irref}<) & \frac{}{x < x} \\
(\text{Tran}<) & \frac{x < y}{x < y, x < z \mid x < y, z < y} && z \text{ is any point variable}
\end{aligned}$$

7.3.3 Axiomatic sets

An axiomatic set is a set including a subset of any of the following forms:

- (a1) $x R y, x -R y$, for either $x, y \in \mathbb{PV}$ and $R \in \mathbb{PRT}$ or $x, y \in \mathbb{IV}$ and $R \in \mathbb{IRT}$;
- (a2) $x = x$ for $x \in \mathbb{PV}$;
- (a3) $x < y, x = y, y < x$ for $x, y \in \mathbb{PV}$;

- (a4) $i U j$ for $i, j \in \mathbb{IV}$;
(a5) $i_1 < i_2, i_1 = i_2$ for $i \in \mathbb{IV}$.

7.3.4 Proof trees and soundness of the proof system

A finite set of formulae $\{x_1 R_1 y_1, \dots, x_n R_n y_n\}$ is said to be an RL_{HS} -set if for every RL_{HS} -model \mathcal{M} and every valuation v in \mathcal{M} there exists $i \in \{1, \dots, n\}$ such that $x_i R_i y_i$ is satisfied by v in \mathcal{M} .

Let Φ be a non-empty set of RL_{HS} -formulae. A rule $\frac{\Phi}{\Phi_1 \mid \dots \mid \Phi_n}$ is RL_{HS} -correct whenever Φ is an RL_{HS} -set if and only if for every $i \in \{1, \dots, n\}$, Φ_i is an RL_{HS} -set. When Φ is empty, RL_{HS} -correctness can be expressed as follows: rule $\frac{}{\Phi_1 \mid \dots \mid \Phi_n}$ is RL_{HS} -correct if and only if there exists $i \in \{1, \dots, n\}$ such that Φ_i is not an RL_{HS} -set.

Definition 7.4. Let $x R y$ be an RL_{HS} -formula. An RL_{HS} -proof tree for $x R y$ is a tree with the following properties:

- the formula $x R y$ is at the root of the tree;
- each node except the root is obtained by application of an RL_{HS} -rule to its predecessor node;
- a node does not have successors whenever it is an RL_{HS} -axiomatic set.

Due to the forms of the rules we obtain the following:

Lemma 7.5. *If a node of an RL_{HS} -proof tree does not contain an axiomatic subset and contains an RL_{HS} -formula $x R y$ or $x -R y$, with R atomic, then all of its successors contain this formula as well.*

A branch of an RL_{HS} -proof tree is said to be *closed* whenever it contains a node with an axiomatic set of formulae. A proof tree is closed if and only if all of its branches are closed.

A formula is *provable* whenever there is a closed RL_{HS} -proof tree for it.

Lemma 7.6.

1. All RL_{HS} -rules are correct.
2. All RL_{HS} -axiomatic sets are RL_{HS} -sets.

Proof. Proof of 1) We show the correctness of rules (B) and $(-E)$. Proving correctness of the other rules is similar. Let $\mathcal{M} = (D, \mathbb{I}(D)^+, m)$ be an RL_{HS} -model and let v be an RL_{HS} -valuation.

It is easy to see that if $\{i B j\}$ is an RL_{HS} -set, then $\{i_1 = j_1, i B j\}$ and $\{j_2 < i_2, i B j\}$ are RL_{HS} -sets. Assume $\mathcal{M}, v \models i_1 = j_1$ and $\mathcal{M}, v \models j_2 < i_2$, that is $v(i), v(j) \in \mathbb{I}(D)^+$, $(v(i_1), v(j_1)) \in m(=)$ and $(v(j_2), v(i_2)) \in m(<)$. By the definition of $m(B)$, we obtain $(v(i), v(j)) \in m(B)$. In the remaining cases the proofs are similar.

The proof of correctness of the rule $(-E)$ is analogous. Assume $\mathcal{M}, v \models i_2 \neq j_2$ or $\mathcal{M}, v \models i_1 \not\prec j_1$, that is $v(i), v(j) \in \mathbb{I}(D)^+$ and $(v(i_2), v(j_2)) \notin m(=)$ or $(v(i_1), v(j_1)) \notin m(<)$. By the definition of $m(E)$, we obtain $(v(i), v(j)) \notin m(E)$, hence $(v(i), v(j)) \in m(-E)$. The remaining parts of the proof are obvious.

Proof of 2) It suffices to show that all sets of the forms (a1)-(a5) are RL_{HS} -sets. We prove it for sets (a4) and (a5). In the remaining cases the proofs are similar.

By the definition of an RL_{HS} -model, for every RL_{HS} -valuation v and for every $i, j \in \mathbb{IV}$, $(v(i), v(j)) \in \mathbb{I}(D)^+ \times \mathbb{I}(D)^+$, hence $(v(i), v(j)) \in m(U)$. Therefore $\{i U j\}$ is an RL_{HS} -set.

By the definition, for every RL_{HS} -valuation v and for every $i \in \mathbb{IV}$, $v(i) \in \mathbb{I}(D)^+$, that is $(v(i_1), v(i_2)) \in m(< \cup =)$. Therefore in every RL_{HS} -model \mathcal{M} , $\mathcal{M}, v \models i_1 < i_2$ or $\mathcal{M}, v \models i_1 = i_2$. Hence $\{i_1 < i_2, i_1 = i_2\}$ is an RL_{HS} -set. \square

Due to Lemma 7.6, we obtain the following theorem.

Theorem 7.7 (Soundness). *Let $x R y$ be an RL_{HS} -formula. If $x R y$ is provable, then it is RL_{HS} -valid.*

7.3.5 Completion conditions

Given a proof tree and a branch b in it, we write, by abusing the notation, $x R y \in b$ if $x R y$ belongs to a set of formulae of a node of branch b . A non-closed branch b is said to be RL_{HS} -complete whenever it satisfies the following *completion conditions*.

For all variables x, y, z and relational terms R, S such that either $x, y, z \in \mathbb{PV}$ and $R, S \in \mathbb{PRT}$, or $x, y, z \in \mathbb{IV}$ and $R, S \in \mathbb{IRT}$:

- $\text{Cpl}(\cup)$ ($\text{Cpl}(-\cap)$) If $x (R \cup S) y \in b$ (resp. $x -(R \cap S) y \in b$), then both $x R y \in b$ (resp. $x -R y \in b$) and $x S y \in b$ (resp. $x -S y \in b$).
- $\text{Cpl}(\cap)$ ($\text{Cpl}(-\cup)$) If $x (R \cap S) y \in b$ (resp. $x -(R \cup S) y \in b$), then $x R y \in b$ (resp. $x -R y \in b$) or $x S y \in b$ (resp. $x -S y \in b$).
- $\text{Cpl}(-)$ If $x (- - R) y \in b$, then $x R y \in b$.
- $\text{Cpl}(-^1)$ If $x R^{-1} y \in b$, then $y R x \in b$.
- $\text{Cpl}(-^{-1})$ If $x -R^{-1} y \in b$, then $y -R x \in b$.
- $\text{Cpl}(:)$ If $x (R; S) y \in b$, then for every z , $x R z \in b$ or $z S y \in b$.
- $\text{Cpl}(-;)$ If $x -(R; S) y \in b$, then for some z both $x -R z \in b$ and $z -S y \in b$.

For all $x, y \in \mathbb{PV}$ and $R \in \mathbb{PRC}$:

- $\text{Cpl}(=1)$ If $x R y \in b$ then, for every $z \in \mathbb{PV}$, $x R z \in b$ or $y = z \in b$.
- $\text{Cpl}(=2)$ If $x R y \in b$ then, for every $z \in \mathbb{PV}$, $z = x \in b$ or $z R y \in b$.

For all $x, y \in \mathbb{P}\mathbb{V}$:

- $\text{Cpl}(\text{Irref} <) \ x < x \in b$.
- $\text{Cpl}(\text{Tran} <) \ \text{If } x < y \in b \text{ then, for every } z \in \mathbb{P}\mathbb{V}, x < z \in b \text{ or } z < y \in b$.

For all $i, j \in \mathbb{I}\mathbb{V}$:

- $\text{Cpl}(R_1) \ \text{If } i R j \in b, \text{ then for every } k \in \mathbb{I}\mathbb{V}, i_1 = k_1 \in b, i_2 = k_2 \in b, \text{ or } k R j \in b$.
- $\text{Cpl}(R_2) \ \text{If } i R j \in b, \text{ then for every } k \in \mathbb{I}\mathbb{V}, j_1 = k_1 \in b, j_2 = k_2 \in b, \text{ or } i R k \in b$.
- $\text{Cpl}(B) \ \text{If } i B j \in b \text{ then } i_1 = j_1 \in b \text{ or } j_2 < i_2 \in b$.
- $\text{Cpl}(-B) \ \text{If } i -B j \in b \text{ then } i_1 \neq j_1, j_2 \not< i_2 \in b$.
- $\text{Cpl}(E) \ \text{If } i E j \in b \text{ then } i_2 = j_2 \in b \text{ or } i_1 < j_1 \in b$.
- $\text{Cpl}(-E) \ \text{If } i -E j \in b \text{ then } i_2 \neq j_2, i_1 \not< j_1 \in b$.

An RL_{HS} -proof tree is said to be RL_{HS} -complete if and only if all of its non-closed branches are RL_{HS} -complete. An RL_{HS} -complete non-closed branch is said to be RL_{HS} -open.

By Lemma 7.5 and since the set containing a subset $\{x R y, x -R y\}$ is axiomatic, the following fact can be easily proved by induction:

Lemma 7.8. *Let b be an open branch of an RL_{HS} -proof tree. Then there is no RL_{HS} -formula $x R y$ such that $x R y \in b$ and $x -R y \in b$.*

7.3.6 Branch model

Let b be an open branch of a proof tree. The branch structure $\mathcal{M}^b = (D^b, \mathbb{I}(D^b)^+, m^b)$ is defined as follows:

- $D^b = \mathbb{P}\mathbb{V}$;
- $m^b(R) = \{(x, y) \in D^b \times D^b : x R y \notin b\}$ for $R \in \text{PRC}$;
- m^b extends to all compound relational terms $R \in \text{PRT}$ as in RL_{HS} -models;
- $\mathbb{I}(D^b)^+ = \{[c, d] : c, d \in D^b, (c, d) \in m^b(< \cup =)\}$;
- $m^b(R) = \{(i, j) \in \mathbb{I}(D^b)^+ \times \mathbb{I}(D^b)^+ : i R j \notin b\}$ for $R \in \text{IRV}$;
- $m^b(U) = \mathbb{I}(D^b)^+ \times \mathbb{I}(D^b)^+$;
- $m^b(B) = \{([c, d], [c', d']) \in \mathbb{I}(D^b)^+ \times \mathbb{I}(D^b)^+ : (c, c') \in m^b(=) \wedge (d', d) \in m^b(<)\}$;
- $m^b(E) = \{([c, d], [c', d']) \in \mathbb{I}(D^b)^+ \times \mathbb{I}(D^b)^+ : (c, c') \in m^b(<) \wedge (d, d') \in m^b(=)\}$;

- m^b extends to all compound relational terms $R \in \mathbb{IRT}$ as in RL_{HS} -models.

Lemma 7.9. $m^b(=)$ is an equivalence relation on D^b .

Proof. $x = x \notin b$ for every $x \in D^b$, because $\{x = x\}$ is axiomatic. Thus, $(x, x) \in m^b(=)$ for every $x \in D^b$. Therefore $m^b(=)$ is reflexive. Assume $(x, y) \in m^b(=)$, that is $x = y \notin b$. Suppose $(y, x) \notin m^b(=)$. Then $y = x \in b$. By the completion condition $\text{Cpl}(=1)$ we get $y = y \in b$ or $x = y \in b$, a contradiction. Therefore $m^b(=)$ is symmetric. Assume $(x, y) \in m^b(=)$ and $(y, z) \in m^b(=)$ that is $x = y \notin b$ and $y = z \notin b$. Suppose $(x, z) \notin m^b(=)$. Then $x = z \in b$. By the completion condition $\text{Cpl}(=1)$ we obtain $x = y \in b$ or $z = y \in b$. In the first case we get a contradiction. In the second case, by the application of the completion condition $\text{Cpl}(=1)$ to $z = y \in b$ we obtain $z = z \in b$ or $y = z \in b$, and in both cases we get a contradiction. Therefore $m^b(=)$ is transitive, and hence $m^b(=)$ is an equivalence relation. \square

Lemma 7.10. Let b be an open branch. A structure \mathcal{M}^b satisfies the conditions (2)-(8) from the definition of RL_{HS} -models.

Proof. The conditions (3)-(8) are satisfied by the definition of a branch structure. Therefore it suffices to show that $m^b(<)$ satisfies the conditions (Irref), (Trans) and (Lin).

By the completion condition $\text{Cpl}(\text{Irref}<)$, for every $x \in D^b$, we have $x < x \in b$, but it means that $(x, x) \notin m^b(<)$ for every $x \in D^b$, therefore $m^b(<)$ is irreflexive.

To prove transitivity, assume $(x, y) \in m^b(<)$ and $(y, z) \in m^b(<)$, that is $x < y \notin b$ and $y < z \notin b$. Suppose $(x, z) \notin m^b(<)$. Then $x < z \in b$. By the completion condition $\text{Cpl}(\text{Tran}<)$ $x < y \in b$ or $y < z \in b$, a contradiction. Therefore $m^b(<)$ satisfies the condition (Trans).

Since b is open, for all $x, y \in D^b$, $x < y \notin b$ or $y < x \notin b$ or $x = y \notin b$. It means that $(x, y) \in m^b(<)$ or $(y, x) \in m^b(<)$ or $(x, y) \in m^b(=)$, therefore $m^b(<)$ satisfies the condition (Lin). \square

Given a structure $\mathcal{M}^b = (D^b, \mathbb{I}(D^b)^+, m^b)$, let $v^b: \mathbb{PV} \cup \mathbb{IV} \rightarrow D^b \cup \mathbb{I}(D^b)^+$ be such that $v^b(x) = x$ for every $x \in \mathbb{PV}$ and $v(i) = [i_1, i_2]$ for every $i \in \mathbb{IV}$.

Lemma 7.11. Let b be an open branch, and let \mathcal{M}^b be the corresponding branch structure. The function v^b satisfies the definition of RL_{HS} -valuation.

Proof. By the definition of v^b , if $x \in \mathbb{PV}$ then $v^b(x) \in D^b$ and, if $i \in \mathbb{IV}$ then $v^b(i) = [v^b(i_1), v^b(i_2)]$. It remains to show that for every $i \in \mathbb{IV}$, $(v^b(i_1), v^b(i_2)) \in m^b(< \cup =)$. Suppose that there exists $i \in \mathbb{IV}$ such that $(v^b(i_1), v^b(i_2)) \notin m^b(< \cup =)$. This implies that $(v^b(i_1), v^b(i_2)) \notin m^b(<)$ and $(v^b(i_1), v^b(i_2)) \notin m^b(=)$. By the definition of m^b , this implies that $i_1 < i_2 \in b$ and $i_1 = i_2 \in b$, which means that b is closed, a contradiction. \square

Satisfiability of formulae in \mathcal{M}^b is defined as for RL_{HS} -models.

Lemma 7.12. *Let b be an open branch and let $x R y$ be an RL_{HS} -formula. Then the following holds:*

$$(*) \text{ if } \mathcal{M}^b, v^b \models x R y, \text{ then } x R y \notin b$$

Proof. The proof is by induction on the complexity of formulae. For $R \in \text{PRC} \cup \text{IRV}$ and its complement, $(*)$ holds by the definition.

- For $R = U$, $(*)$ holds trivially, since $i U j$ is axiomatic.
- Let $R = B$. Assume $(i, j) \in m^b(B)$. This implies that $(i_1, j_1) \in m^b(=)$ and $(j_2, i_2) \in m^b(<)$. Then $i_1 = j_1 \notin b$ and $j_2 < i_2 \notin b$. Suppose $i B j \in b$. By the completion condition $\text{Cpl}(B)$, $i_1 = j_1 \in b$ or $j_2 < i_2 \in b$, a contradiction.
- Let $R = -B$. Assume $(i, j) \notin m^b(B)$. This implies that $(i_1, j_1) \notin m^b(=)$ or $(j_2, i_2) \notin m^b(<)$. Then $i_1 = j_1 \in b$ or $j_2 < i_2 \in b$. Suppose $i -B j \in b$. By the completion condition $\text{Cpl}(-B)$, both $i_1 \neq j_1 \in b$ and $j_2 \not< i_2 \in b$, a contradiction.
- Let $R = E$. Assume $(i, j) \in m^b(E)$. This implies that $(i_1, j_1) \in m^b(<)$ and $(j_2, i_2) \in m^b(=)$. Then $i_1 < j_1 \notin b$ and $j_2 = i_2 \notin b$. Suppose $i E j \in b$. By the completion condition $\text{Cpl}(E)$, $i_1 < j_1 \in b$ or $j_2 = i_2 \in b$, a contradiction.
- Let $R = -E$. Assume $(i, j) \notin m^b(E)$. This implies that $(i_1, j_1) \notin m^b(<)$ or $(j_2, i_2) \notin m^b(=)$. Then $i_1 < j_1 \in b$ or $j_2 = i_2 \in b$. Suppose $i -E j \in b$. By the completion condition $\text{Cpl}(-E)$, both $i_1 \not< j_1 \in b$ and $j_2 \neq i_2 \in b$, a contradiction.

Therefore $(*)$ holds for all atomic formulae and their complements. The remaining cases can be proved in a standard way using the completion conditions and the property of Lemma 7.8. See also [GPO06a]. \square

It is easy to check that the branch structure satisfies the following extensionality property.

Lemma 7.13. *Let \mathcal{M}^b be a branch structure determined by an open branch b . Then the following hold:*

- For every $R \in \text{PRC}$ and for all $x, y, z, t \in \text{PV}$ we have that if $(x, y) \in m^b(R)$ and $(x, z), (y, t) \in m^b(=)$, then $(z, t) \in m^b(R)$.
- For every $R \in \text{IRA}$ and for all $i, j, k, l \in \text{IV}$ such that $i = [i_1, i_2], j = [j_1, j_2], k = [k_1, k_2], l = [l_1, l_2]$: if $(i, j) \in m^b(R)$ and $(i_1, k_1), (i_2, k_2), (j_1, l_1), (j_2, l_2) \in m^b(=)$, then $(k, l) \in m^b(R)$.

Since $m^b(=)$ is an equivalence relation on D^b , given a branch structure \mathcal{M}^b , we may define the quotient model $\mathcal{M}_q^b = (D_q^b, \mathbb{I}(D_q^b)^+, m_q^b)$ as follows:

- $D_q^b = \{\|x\| : x \in D^b\}$, where $\|x\|$ is the equivalence class of $m^b(=)$ generated by x ;

- $\mathbb{I}(D_q^b)^+ = \{[\|c\|, \|d\|] : [c, d] \in \mathbb{I}(D^b)^+\}$;
- $m_q^b(R) = \{(\|x\|, \|y\|) \in D_q^b \times D_q^b : (x, y) \in m^b(R)\}$, for every $R \in \text{PRC}$;
- m_q^b extends to all compound relational terms $R \in \text{PRT}$ as in RL_{HS} -models;
- $m_q^b(R) = \{([\|c\|, \|d\|], [\|c'\|, \|d'\|]) \in \mathbb{I}(D_q^b)^+ \times \mathbb{I}(D_q^b)^+ : ([c, d], [c', d']) \in m^b(R)\}$, for every $R \in \text{IRA}$;
- m_q^b extends to all compound relational terms $R \in \text{IRT}$ as in RL_{HS} -models.

Due to Lemma 7.13, the quotient model \mathcal{M}_q^b is well defined, that is the definitions of $m_q^b(R)$ and $\mathbb{I}(D_q^b)^+$ do not depend on the choice of the representatives of the equivalence classes.

Lemma 7.14. *The structure \mathcal{M}_q^b is an RL_{HS} -model.*

Proof. We have to show that $m_q^b(=)$ is the identity on D_q^b . Indeed, for every $x, y \in \mathbb{PV}$ we have:

$$(\|x\|, \|y\|) \in m_q^b(=) \text{ iff } (x, y) \in m^b(=) \text{ iff } \|x\| = \|y\|$$

□

Let v_q^b be such that $v_q^b(x) = \|x\|$, for every $x \in \mathbb{PV}$, and $v_q^b(i) = [\|i_1\|, \|i_2\|]$, for every $i \in \text{IV}$. It is easy to see that v_q^b is an RL_{HS} -valuation in \mathcal{M}_q^b , since v^b satisfies the definition of RL_{HS} -valuation.

By an easy induction we can prove the following:

Lemma 7.15. *For every RL_{HS} -formula $x R y$:*

$$(*) \quad \mathcal{M}^b, v^b \models x R y \quad \text{iff} \quad \mathcal{M}_q^b, v_q^b \models x R y$$

The above lemmas enable us to prove the completeness of the proof system.

Theorem 7.16 (Completeness of RL_{HS} -system). *Let $x R y$ be an RL_{HS} -formula. If $x R y$ is RL_{HS} -valid, then $x R y$ is RL_{HS} -provable.*

Proof. Assume $x R y$ is RL_{HS} -valid. Suppose there is no closed RL_{HS} -proof tree for $x R y$. Consider a non-closed RL_{HS} -proof tree for $x R y$. We may assume that this tree is complete. Let b be an open branch of the complete RL_{HS} -proof tree for $x R y$. Since $x R y \in b$, by Lemma 7.12, the branch structure \mathcal{M}^b does not satisfy $x R y$. By Lemma 7.15 also the quotient model \mathcal{M}_q^b does not satisfy $x R y$. Since \mathcal{M}_q^b is an RL_{HS} -model, $x R y$ is not RL_{HS} -valid, a contradiction. □

7.4 HS-validity and RL_{HS} -provability

In this section we conclude the discussion of Sections 7.2 and 7.3 and we show how the proof system of logic RL_{HS} can be used to verify the validity and entailment of formulae of logic HS. We also present examples of derivations.

The following theorem follows from Theorems 7.3 and 7.16.

Theorem 7.17. *For every HS-formula φ , φ is HS-valid if and only if $i \tau(\varphi) j$ is RL_{HS} -provable.*

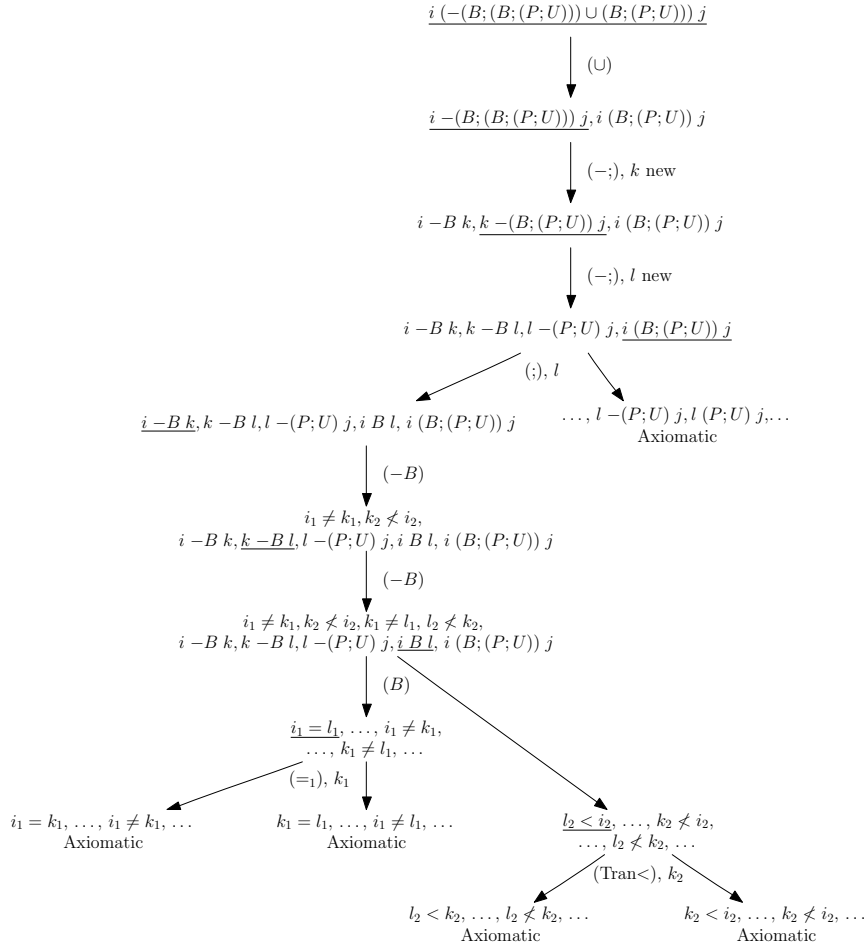


Figure 7.1: Proof tree for $\langle B \rangle \langle B \rangle p \rightarrow \langle B \rangle p$

As an example of validity checking, consider the HS-formula $\varphi = \langle B \rangle \langle B \rangle p \rightarrow \langle B \rangle p$, which express the fact that $\langle B \rangle$ is a transitive modality. By the semantics of HS, it is

easy to see that φ is valid. The translation $\tau(\varphi)$ of the above formula into a relational term of RL_{HS} is $\underline{-(B; (B; (P; U)))} \cup (B; (P; U))$. Figure 7.1 depicts an RL_{HS} -proof tree that shows that the relational formula $i \tau(\varphi) j$ is RL_{HS} -valid, and thus that φ is HS-valid. In each node of the proof tree we underline the formula to which a rule has been applied during the construction of the proof tree.

Let R_1, \dots, R_n, R be binary relations on $\mathbb{I}(D)^+$ and let $U = \mathbb{I}(D)^+ \times \mathbb{I}(D)^+$. It is known that $R_1 = U, \dots, R_n = U$ imply $R = U$ iff $(U; -(R_1 \cap \dots \cap R_n); U) \cup R = U$. Therefore, for every RL_{HS} -model \mathcal{M} , $\mathcal{M} \models i R_1 j, \dots, \mathcal{M} \models i R_n j$ imply $\mathcal{M} \models i R j$ iff $\mathcal{M} \models i (U; -(R_1 \cap \dots \cap R_n); U) \cup R j$ which means that entailment in RL_{HS} can be expressed in its language.

As an example of entailment in RL_{HS} , suppose that $<$ is a dense linear ordering. It can be shown that density can be expressed in terms of the relation B by the following axiom:

$$\text{Dense}_{\text{RL}_{HS}} := B \subseteq (B; B),$$

that is equivalent to $\underline{-B} \cup (B; B) = U$. In [Ven90], the following HS-axiom is proposed to express density:

$$\text{Dense}_{HS} := \langle \overline{B} \rangle p \rightarrow \langle \overline{B} \rangle \langle \overline{B} \rangle p.$$

Its RL_{HS} -translation is $\tau(\text{Dense}_{HS}) = \underline{-(B^{-1}; (P; U))} \cup (B^{-1}; (B^{-1}; (P; U)))$. To prove that $\text{Dense}_{\text{RL}_{HS}}$ entails Dense_{HS} it is sufficient to show that the relational formula

$$i [(U; \underline{-(-B \cup (B; B))}; U) \cup \underline{-(B^{-1}; (P; U))} \cup (B^{-1}; (B^{-1}; (P; U)))] j$$

is RL_{HS} -valid. Figure 7.2 depicts a closed proof tree for this formula, thus proving that Dense_{HS} is valid for every dense ordering. As in the previous example, in each node of the proof tree we underline the formula to which a rule has been applied during the construction of the proof tree.

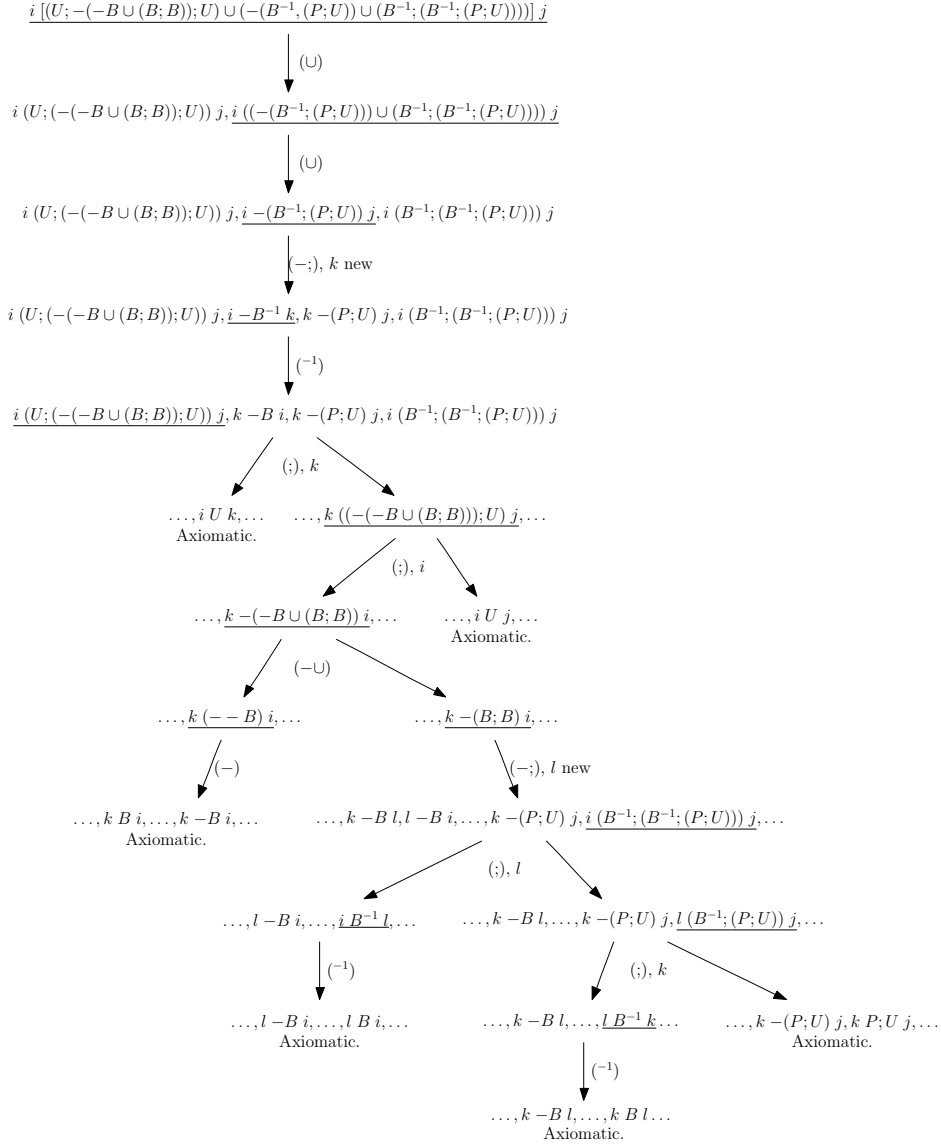
7.5 Extensions of the relational system

In the previous sections we have provided a relational proof system for the interval temporal logic HS, interpreted over linear temporal domains. In this section we exploit the modularity of the relational approach, and we show how to adapt it to cope with other interval relations and other meaningful temporal domains.

7.5.1 Incorporating the other interval relations

In this section we show how to modify the relational logic RL_{HS} and its proof system to obtain a relational logic RL_L (and a corresponding proof system) that is appropriate to any interval logic L that is based on unary modalities corresponding to Allen's relations. As discussed in Chapter 1, any interval logic L based on Allen's relations can be defined by choosing a subset of the following temporal operators:

$$T = \{\pi, \langle E \rangle, \langle \overline{E} \rangle, \langle D \rangle, \langle \overline{D} \rangle, \langle B \rangle, \langle \overline{B} \rangle, \langle O \rangle, \langle \overline{O} \rangle, \langle A \rangle, \langle \overline{A} \rangle, \langle F \rangle, \langle \overline{F} \rangle\}.$$

Figure 7.2: Proof tree showing that $\text{Dense}_{\text{RL}_{\text{HS}}}$ entails Dense_{HS}

Given an interval logic L , the corresponding relational logic RL_L differs from RL_{HS} only in the choice of the set of interval relational constants, that is defined as $\mathbb{IRC} = \{U\} \cup \{R : \langle R \rangle \in L\} \cup \{\Pi : \pi \in L\}$. Models of RL_L are defined as in the case of RL_{HS} while the semantics of the relational constants R has to be defined in accordance with the semantics of the chosen primitive interval relations. Any L -formula φ can be translated to an RL_L -formula $i R j$ by means of the following validity preserving translation τ :

- for propositional letters and for propositional connectives, τ is defined as in the case of RL_{HS} ;
- if the modal constant π is in the language, $\tau(\pi) = \Pi; U$;
- for every basic modality $\langle R \rangle$, $\tau(\langle R \rangle \psi) = R; \tau(\psi)$;
- for every converse modality $\langle \bar{R} \rangle$, $\tau(\langle \bar{R} \rangle \psi) = R^{-1}; \tau(\psi)$.

A proof system for RL_L can be obtained from the proof system for RL_{HS} by substituting rules (B) , (E) , $(-B)$, and $(-E)$ with rules that are appropriate for the choice of basic Allen's relations. Rules for *begins* and *ends* are presented in Section 7.3, while the rules for the remaining relations are the following.

For $i, j \in \mathbb{IV}$:

$$\begin{array}{ll}
(\Pi) \frac{i \Pi j}{i_1 = i_2, i \Pi j} & (-\Pi) \frac{i -\Pi j}{i_1 \neq i_2, i -\Pi j} \\
(D) \frac{i D j}{i_1 < j_1, i D j \mid j_2 < i_2, i D j} & (-D) \frac{i -D j}{i_1 \not< j_1, j_2 \not< i_2, i -D j} \\
(A) \frac{i M j}{i_2 = j_1, i M j} & (-A) \frac{i -M j}{i_2 \neq j_1, i -M j} \\
(F) \frac{i P j}{i_2 < j_1, i P j} & (-F) \frac{i -P j}{i_2 \not< j_1, i -P j} \\
(O) \frac{i O j}{j_1 < i_1, i O j \mid i_1 < j_2, i O j \mid j_2 < i_2, i O j} & \\
(-O) \frac{i -O j}{j_1 \not< i_1, i_1 < j_2, j_2 < i_2, i -O j} &
\end{array}$$

It is easy to check that the rules correspond to the semantics of Allen's relations, as defined in Chapter 1. Hence, soundness of the rules is straightforward. To prove completeness we need to appropriately expand the completion conditions and the notion of branch structure. For instance, rules (A) and $(-A)$ require the following completion conditions:

$$\begin{array}{ll}
\text{Cpl}(M) & \text{If } i M j \in b \text{ then } i_2 = j_1 \in b. \\
\text{Cpl}(-M) & \text{If } i -M j \in b \text{ then } i_2 \neq j_1 \in b.
\end{array}$$

Consider now the branch structure $\mathcal{M}^b = (D^b, \mathbb{I}(D^b)^+, m^b)$. The meaning of A in \mathcal{M}^b is defined as follows:

$$m^b(A) = \{([c, d], [c', d']) \in \mathbb{I}(D^b) \times \mathbb{I}(D^b) : (d, c') \in m^b(=)\}.$$

The valuation v^b and the notion of satisfiability in \mathcal{M}^b are defined as in RL_{HS} . To prove completeness, we have to show that $\mathcal{M}^b, v^b \models i R j$ if and only if $i R j \notin b$, where R can be either A or $\neg A$.

- Let $R := A$. Assume $(i, j) \in m^b(A)$, that is $(i_2, j_1) \in m^b(=)$. Then $i_2 = j_1 \notin b$. Suppose $i A j \in b$. By the completion condition $\text{Cpl}(A)$, $i_2 = j_1 \in b$, a contradiction.
- Let $R := \neg A$. Assume $(i, j) \notin m^b(A)$, that is $(i_2, j_1) \notin m^b(=)$. Then $i_2 = j_1 \in b$. Suppose $i \neg A j \in b$. By the completion condition $\text{Cpl}(\neg A)$, $i_2 \neq j_1 \in b$, a contradiction.

The remaining part of the completeness proof is as in RL_{HS} .

Relational systems for other interval temporal logics.

The rules presented above allow us to easily adapt the proof system for RL_{HS} to any propositional interval temporal logic that is a proper fragment of HS. Here we show two examples of such a modification.

The logic BE. The logic BE features the two modalities $\langle B \rangle$ and $\langle E \rangle$, and was first studied in [Lod00], where its undecidability has been proved. Since BE does not have converse modalities, the relational logic RL_{BE} appropriate for BE is logic RL_{HS} without the converse operator $^{-1}$. A relational proof system for RL_{BE} can be obtained from the one for RL_{HS} by removing rules $(^{-1})$ and (\neg^{-1}) .

Propositional neighborhood logics. The relational logic $\text{RL}_{\text{PNL}^{\pi+}}$ (appropriate for $\text{PNL}^{\pi+}$) is logic RL_{HS} where the interval relational constant Π and A takes place of B and E . A proof system for $\text{RL}_{\text{PNL}^{\pi+}}$ can be obtained from the one for RL_{HS} by substituting rules (B) , $(\neg B)$, (E) , and $(\neg E)$ with rules (Π) , $(\neg \Pi)$, (A) and $(\neg A)$.

7.5.2 Considerations on the nature of intervals

In this chapter we always considered the *non-strict* semantics for interval logics. As shown in Chapter 1, another natural semantics for interval logics is considered, namely, the *strict* one, where point intervals are excluded.

Given a relational proof system RL_L for an interval logic L , we show how to modify it in the case of the strict semantics. To this end, we define the relational logic RL_L^- (strict RL_L), characterized by the same syntax as non-strict RL_L , but with a different semantics. An RL_L^- -model is a tuple $\mathcal{M}^- = (D, \mathbb{I}(D)^-, m)$ where D and m are defined as in RL_L -models, and $\mathbb{I}(D)^- = \{[c, d] \in D \times D : (c, d) \in m(<)\}$. An RL_L^- -valuation is any function $v: \mathbb{P}V \cup \mathbb{I}V \rightarrow 2^D \cup 2^{\mathbb{I}(D)^- \times \mathbb{I}(D)^-}$ such that:

- if $x \in \mathbb{P}\mathbb{V}$ then $v(x) \in D$;
- if $i \in \mathbb{I}\mathbb{V}$ then $v(i) = [v(i_1), v(i_2)] \in \mathbb{I}(D)^-$.

The notions of satisfiability and validity of a formula are defined as in RL_L .

A proof system for RL_L^-

A proof system for RL_L^- can be obtained from the proof system for RL_L by substituting the axiomatic set (a5) with a new one:

$$(a5^-) \quad i_1 < i_2 \text{ for } i \in \mathbb{I}\mathbb{V}.$$

In the case of the strict semantics, for every valuation v and every interval variable i , we have $v(i) = [v(i_1), v(i_2)]$ with $v(i_1) < v(i_2)$. Hence, (a5⁻) is an RL_L^- -set. Correctness of the other rules of the proof system follows directly from the correctness of the rules for RL_L . Thus, soundness of the RL_L^- -proof system is straightforward.

Completeness of the proof system can be proved as in the case of RL_L , with the only difference that, given an open branch b , the branch structure $\mathcal{M}^b = (D^b, \mathbb{I}(D^b)^-, m^b)$ is defined such that $\mathbb{I}(D^b)^- = \{[c, d] : c, d \in D^b, (c, d) \in m^b(<)\}$.

7.5.3 Properties of the temporal ordering

In all the relational systems RL_L presented above, the strict ordering $<$ is considered to be linear, without any further assumption. In this section we propose some possible extensions and modifications of our systems in case of other temporal orderings.

Unbounded orderings

An ordering is said to be *unboundend below* (resp. *above*) if for every x there exists z such that $z < x$ (resp. $x < z$). Such a condition can be expressed in a relational system RL_L by means of the following rules.

For $x \in \mathbb{P}\mathbb{V}$:

$$(\text{No-min}<) \quad \frac{}{z \not< x} \quad (\text{No-max}<) \quad \frac{}{x \not< z} \quad \text{with } z \text{ new point variable.}$$

Soundness of the rules can be easily proved. Suppose that $<$ is unbounded below (the case where $<$ is unbounded above is similar). Then, for every x , there exists z such that $z < x$. Thus, $z \not< x$ cannot be an RL_L -set and rule (No-min $<$) is correct.

To prove completeness of the system, we need to add the following completion conditions.

$$\text{Cpl}(\text{No-min}<) \quad \text{For all } x \in \mathbb{P}\mathbb{V}, \text{ there exists } z \in \mathbb{P}\mathbb{V} \text{ such that } z \not< x \in b.$$

$$\text{Cpl}(\text{No-max}<) \quad \text{For all } x \in \mathbb{P}\mathbb{V}, \text{ there exists } z \in \mathbb{P}\mathbb{V} \text{ such that } x \not< z \in b.$$

Consider now the branch structure $\mathcal{M}^b = (D^b, \mathbb{I}(D^b), m^b)$ of RL_L . To prove that $m^b(<)$ is unbounded below, suppose by contradiction that there exists $x \in \mathbb{P}\mathbb{V}$ such that, for all $z \in \mathbb{P}\mathbb{V}$, $(z, x) \notin m^b(<)$. This implies that $z < x \in b$ for all $z \in \mathbb{P}\mathbb{V}$. By

the completion condition $\text{Cpl}(\text{No-min}<)$, there exists $z \in \mathbb{P}\mathbb{V}$ such that $z \not\prec x \in b$ and $z < x \in b$, a contradiction. Proving that the completion condition $\text{Cpl}(\text{No-max}<)$ implies that $m^b(<)$ is unbounded above is similar.

Dense orderings

An ordering $<$ is *dense* if for every pair of different comparable points there exists another point in between, namely, if $\forall x, y(x < y \rightarrow \exists z(x < z \wedge z < y))$ holds. Density of the time domain can be expressed by the following rule.

For $x, y \in \mathbb{P}\mathbb{V}$:

$$(\text{Dense}<) \quad \frac{}{x < y \mid x \not\prec z, z \not\prec y} \quad \text{with } z \text{ new point variable.}$$

Soundness is straightforward: the rule corresponds to the first-order formula $\exists x, y(x < y \wedge \forall z(x \not\prec z \vee z \not\prec y))$, that is exactly the negation of the density condition. As for the completeness, we add the following completion condition.

$\text{Cpl}(\text{Dense}<)$ For all $x, y \in \mathbb{P}\mathbb{V}$, $x < y \in b$ or there exists $z \in \mathbb{P}\mathbb{V}$ such that $x \not\prec z \in b$ and $z \not\prec y \in b$.

Consider now the branch structure, and suppose that $m^b(<)$ does not respect the density condition, that is, there exist $x, y \in \mathbb{P}\mathbb{V}$ such that $(x, y) \in m^b(<)$ and, for all $z \in \mathbb{P}\mathbb{V}$, $(x, z) \notin m^b(<)$ or $(z, y) \notin m^b(<)$. This implies that $x < y \notin b$ and, for all z , $x < z \in b$ or $z < y \in b$. By the completion condition $\text{Cpl}(\text{Dense}<)$, we have that there exists z such that $x \not\prec z \in b$ and $z \not\prec y \in b$, a contradiction.

Discrete orderings

An ordering is *discrete* if every point with a successor/predecessor has an immediate successor/predecessor, that is:

$$(1) \quad \forall x, y(x < y \rightarrow \exists z(x < z \wedge \forall t(x \not\prec t \vee t \not\prec z))),$$

and

$$(2) \quad \forall x, y(y < x \rightarrow \exists z(z < x \wedge \forall t(z \not\prec t \vee t \not\prec x))).$$

Discreteness of the time domain is expressed by the following additional rules.

For $x, y, z, t \in \mathbb{P}\mathbb{V}$:

$$(\text{Disc}<_1) \quad \frac{}{x < y \mid x \not\prec z, x < t \mid x \not\prec z, t < z}$$

$$(\text{Disc}<_2) \quad \frac{}{y < x \mid z \not\prec x, z < t \mid z \not\prec x, t < x}$$

with x, y, t any point variable, z new point variable.

The lower part of rule $(\text{Disc}<_1)$ corresponds to the first-order formula $\exists x, y(x < y \wedge \forall z(x \not\prec z \vee \exists t(x < t \wedge t < z)))$, that is exactly the negation of condition (1). Similarly, the lower part of rule $(\text{Disc}<_2)$ corresponds to the negation of condition (2). Hence, soundness of the rules is straightforward.

To prove completeness, it is necessary to add the following completion conditions to the system:

Cpl(Disc_{<1}) For all $x, y \in \mathbb{P}\mathbb{V}$, $x < y \in b$, or there exists $z \in \mathbb{P}\mathbb{V}$ such that $x \not< z \in b$ and, for all $t \in \mathbb{P}\mathbb{V}$, $x < t \in b$, or $t < z \in b$.

Cpl(Disc_{<2}) For all $x, y \in \mathbb{P}\mathbb{V}$, $y < x \in b$, or there exists $z \in \mathbb{P}\mathbb{V}$ such that $z \not< x \in b$ and, for all $t \in \mathbb{P}\mathbb{V}$, $z < t \in b$, or $t < x \in b$.

Consider now the branch structure \mathcal{M}^b , and suppose that $m^b(<)$ does not respect condition (1). This implies that there exist $x, y \in \mathbb{P}\mathbb{V}$ such that $(x, y) \in m^b(<)$ but, for all $z \in \mathbb{P}\mathbb{V}$, $(x, z) \notin m^b(<)$, or there exists $t \in \mathbb{P}\mathbb{V}$ such that $(x, t) \in m^b(<)$ and $(t, z) \in m^b(<)$. By the definition of branch structure, this implies that $x < y \notin b$ and $x < z \in b$ or $x < t \notin b$ and $t < z \notin b$. By the completion condition Cpl(Disc_{<1}), one of the following may arise:

- $x < y \in b$, a contradiction;
- $x \not< z \in b$ and $x < t \in b$, a contradiction;
- $x \not< z \in b$ and $t < z \in b$, a contradiction.

Conclusions

In this dissertation we have studied and developed tableau and dual-tableau based proof systems for propositional interval temporal logics. We first focused our attention on propositional neighborhood logics, by providing tableau-based decision procedure for them and by discussing their decidability and expressivity. Then we developed a relational dual-tableau that can be used as a proof system for most interval temporal logics with unary modalities.

Chapters 3, 4, 5, and 6 are devoted studying to the decidability and expressivity of various propositional neighborhood logics. We started, in Chapter 3, by focusing our attention on the future fragment of Propositional Neighborhood Logic (RPNL) interpreted over natural numbers. We addressed the satisfiability problem for RPNL, showing 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^{π^+} and we proved its optimality. We concluded the chapter by briefly showing that such a procedure can be easily adapted to RPNL^+ and RPNL^- . The proposed decision procedure improves the EXPSPACE tableau-based decision method for checking RPNL^- satisfiability developed by Bresolin and Montanari in [BM05b] and it is a generalization to RPNL^{π^+} of the tableau-based decision procedure originally developed for RPNL^- in [BMS07b].

In Chapter 4 we generalized this method to the branching-time propositional interval temporal logic BTNL. We recall from Chapter 1 that such a logic is interpreted over infinite trees and it combines the interval neighborhood operators of RPNL with the path quantifiers of CTL. By combining Emerson and Halpern's tableau-based decision procedure for CTL [EH85] with the one we developed in this chapter for RPNL, we have been able to devise a doubly-exponential tableau-based decision procedure for BTNL.

The extension of the tableau method for RPNL to full PNL turned out to be difficult. In particular, there is not a straightforward way of generalizing the basic removal technique exploited in Chapter 3 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* it may introduce. Chapter 5 is devoted to show how these problems can be solved, in order to obtain a tableau-based decision procedure for full PNL interpreted over the integers and over subsets of them, such as the naturals or finite linear orderings. In particular, we have developed a NEXPTIME tableau-based decision procedure for PNL^{π^+} that can be adapted to PNL^- and PNL^+ as well. Thanks to the NEXPTIME-hardness of PNL, such a decision procedure is of optimal complexity.

Finally, in Chapter 6 we explored expressiveness and decidability issues for Propositional Neighborhood Logics. First, we compared PNL^{π^+} with PNL^+ and PNL^- ,

and we showed that the former is strictly more expressive than the other two. Then, we proved that $\text{PNL}^{\pi+}$ is decidable by embedding it into $\text{FO}^2[<]$. Next, we proved that $\text{PNL}^{\pi+}$ is as expressive as $\text{FO}^2[<]$. Finally, we compared $\text{PNL}^{\pi+}$ with HS and other interval logics.

The results of Chapters 3, 4, 5, and 6 can be further developed in several directions.

- In the case of BTNL, we do not know yet if the satisfiability problem for this logic is doubly EXPTIME-complete or not. In our tableau method nodes of the tableau are defined as sets of atoms, thus giving a doubly-exponential blow-up in the tableau construction. We conjecture that the complexity of the satisfiability problem for BTNL can be lowered by devising a tableau method where nodes are *single* atoms (like in the tableau for RPNL and CTL), thus avoiding the doubly-exponential state explosion of our current method.
- Another possible extension of the tableau method for BTNL is its generalization to the decision problem for logics that combine path quantifiers operators with other sets of interval logic operators, e.g., those of PNL or of other propositional interval logics.
- The tableau methods we give for PNL and RPNL can be optimized by exploiting an on-the-fly approach for the tableau construction. Given a formula φ , our methods first construct the closure of φ and the set of φ -atoms, and then apply rules that check for satisfiability. This immediately causes an exponential blowup, which is necessary in the worst case, but may not be necessary in the average case. By contrast, with an on-the-fly approach the procedure simply breaks down the formula φ into smaller and smaller sets of formulae, from which a model can be built, or a contradiction found, as soon as is possible. The worst-case complexity of the method remains the same, but it is likely that the exponential blowup can be avoided in most cases.
- A second possibility is to explore the extension of the tableau for RPNL and PNL to temporal domains different from the naturals and the integers. In Chapter 6 we give a non-constructive proof of the decidability of PNL over the class of all linear orderings and other orderings such as the class of all Dedekind-complete linear orderings. The development of an effective decision procedure for these cases is an extension that it is worth exploring.
- There are some interesting classes of orderings, like dense linear orderings, the reals, and partial orderings with the linear interval property, where the decidability of PNL is still unknown. It would be of great interest to explore the possibility of extending the results of Chapter 6 to these classes of orderings.
- One could imagine extending PNL with other interval modalities, such as the subinterval modalities $\langle D \rangle$ and $\langle \overline{D} \rangle$ or, in the case of the integers, to modalities corresponding to the successor/predecessor relations, and to devise a tableau-based decision procedure for such new logics, if they turn out to be decidable,

or to prove their undecidability. This would extend the known results about decidability and undecidability in interval temporal logics, possibly by adding new, expressive, but still decidable, interval logics to the ones that have been studied so far.

- As for expressiveness, in this dissertation we partially explored the relationships between PNL and other fragment of HS. A comparison with point-based temporal logics can be of interest as well. For example, there is an obvious embedding of the standard point-based temporal logic $TL[F,P]$ into $PNL^{\pi+}$. The (non-)existence of the opposite embedding is more interesting, but also more difficult to state in a precise way.

In Chapter 7 we presented a sound and complete relational proof system for Halpern and Shoham's HS logic interpreted over non-strict linear interval structures. Furthermore, we showed how to extend the system to the class of all interval temporal logics based on Allen's relations, interpreted over strict and non-strict interval structures and over linear orderings with various specific properties (e.g., unbounded, dense, discrete).

The relational approach presented in Chapter 7 can be extended in several ways.

- A first possibility is the development of a relational logic and of a relational proof system that are adequate for interval logics based on binary modalities, like CDT. Such an extension must include relational constants corresponding to the C , D , and T modalities of CDT, while the rules of the proof system must reflect their semantics. An example of a relational approach to a modal logic with binary modalities can be found in [Orl95], where the *Since* and *Until* operators of point-based temporal logics has been considered.
- A semantic extension of the method is its generalization to interval structures based on partial orderings with the linear interval property (see Chapter 1). In this case, the strict ordering $<$ is not a linear ordering any more. It is transitive and irreflexive, but the linearity constraint is substituted with the linear interval property. Thus, the axiomatic set (a3) presented in Section 7.3.3 is not correct anymore and should be substituted with new rules and/or new axiomatic sets that are correct with respect to partial orderings with the linear interval property.
- A more interesting, yet more difficult development of our relational approach is the use of dual tableau as decision procedures for (decidable) interval logics. Since HS is an undecidable logic, the relational proof system for RL_{HS} proposed in this chapter is a semi-decision procedure. Even when restricted to interval logics that are known to be decidable, it remains a non-terminating procedure. It is not clear how relational proof systems can be turned into decision procedures, when the considered logic is decidable. Since tableaux and dual tableaux are known to be dual in a precisely defined sense [GPO06b], it would be interesting to explore this relationship in order to derive a terminating dual tableau

for interval logics for which a terminating tableau-based decision procedure is available, like Bowman and Thompson's local PITL [BT03], or the propositional neighborhood logics RPNL, PNL and BTNL.

Finally, we conclude the dissertation by outlining some open problems in the area of interval logics that are not directly connected with the results presented here.

- The subinterval logic D is the fragment of HS featuring only the subinterval modality $\langle D \rangle$. As pointed out in Chapter 1, there are few results in the literature about decidability and axiomatizability of this logic when over some specific structures (such as dense linear ordering). Decidability and axiomatizability of the D logic in the general case is still an open problem.
- Another fragment of HS that is worth studying is the logic $\langle D \rangle / \langle \overline{D} \rangle$, featuring the subinterval and superinterval modalities. In [Lod00], Lodaya conjectures that this logic is undecidable. However, we are not aware of any published results about its decidability or undecidability.
- In this dissertation we mostly focused on the satisfiability problem for interval logics. Another interesting problem, that has been extensively studied for point-based temporal logics, is the model checking problem. There are very few published works about model checking for interval temporal logics [CCG00, PPH98], despite the great interest for such topics in computer science.

List of Figures

1.1	Allen's relations	4
1.2	Venema's ternary relation A	5
1.3	The split structure of the naturals.	11
2.1	An example of the tableau method for LPITL_{proj}	20
2.2	An example of the tableau method for BCDT^+	23
3.1	The layered structure	28
3.2	A tableau for $\Box_r \pi$	44
4.1	Connecting two nodes.	52
4.2	A portion of the tableau for $E\Box_r p$	58
6.1	Expressive completeness results for interval logics.	87
7.1	Proof tree for $\langle B \rangle \langle B \rangle p \rightarrow \langle B \rangle p$	102
7.2	Proof tree showing that $\text{Dense}_{\text{RLHS}}$ entails Dense_{HS}	104

Bibliography

- [All83] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- [BGG97] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives of Mathematical Logic. Springer, 1997.
- [BGMS07] 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.
- [BGPO06] D. Bresolin, J. Golińska-Pilarek, and E. Orłowska. Relational dual tableaux for interval temporal logics. *Journal of Applied Non-Classical Logics*, 16(3–4):251–277, 2006.
- [BM05a] D. Bresolin and A. Montanari. A tableau-based decision procedure for a branching-time interval temporal logic. In H. Schlingloff, editor, *Proceedings of M4M-4: 4th International Workshop on Methods for Modalities*, pages 38–53, Berlin, Germany, December 2005.
- [BM05b] 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, Koblenz, Germany, September 2005. Springer.
- [BMS07a] D. Bresolin, A. Montanari, and P. Sala. An optimal tableau-based decision algorithm for Propositional Neighborhood Logic. In *Proc. of STACS 2007: 24th International Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *LNCS*, pages 549–560, Aachen, Germany, February 2007.
- [BMS07b] 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.
- [BT03] 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.
- [CCG00] S. Campos, E. M. Clarke, and O. Grumberg. Selective quantitative analysis and interval model checking: Verifying different facets of a system. *Formal Methods in System Design*, 17(2):163–192, 2000.

- [CH98] Z. Chaochen and M. R. Hansen. An Adequate First Order Interval Logic. In W. de Roever, H. Langmark, and A. Pnueli, editors, *Compositionality: the Significant Difference*, volume 1536 of *LNCS*, pages 584–608. Springer, 1998.
- [CR03] A. V. Chagrov and M. N. Rybakov. How Many Variables Does One Need to Prove PSPACE-hardness of Modal Logics. In P. Balbiani, Nobu-Yuki Suzuki, F. Wolter, and M. Zakharyashev, editors, *Advances in Modal Logic*, volume 4, pages 71–82. King’s College Publications, 2003.
- [CSFDC00] N. Chetcuti-Serandio and L. Fariñas Del Cerro. A mixed decision method for duration calculus. *Journal of Logic and Computation*, 10:877–895, 2000.
- [DG00] S. Demri and R. Goré. An $O((n \cdot \log n)^3)$ -Time Transformation from Grz into Decidable Fragments of Classical First-Order Logic. In *Automated Deduction in Classical and Non-Classical Logics*, volume 1761 of *LNAI*, pages 153–167. Springer, 2000.
- [DGHP99] M. D’Agostino, D. Gabbay, R. Hänle, and J. Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Press, 1999.
- [DOR94] S. Demri, E. Orłowska, and I. Rewitzky. Towards reasoning about Hoare relations. *Annals of Mathematics and Artificial Intelligence*, 12:265–289, 1994.
- [EH85] 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.
- [Eme90] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 995–1072. MIT Press, 1990.
- [EVW02] 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.
- [Fit83] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169. D. Reidel, Dordrecht, Holland, 1983.
- [FO95] M. Frias and E. Orłowska. A proof system for fork algebras and its applications to reasoning in logics based on intuitionism. *Logique et Analyse*, 150–151–152:239–284, 1995.
- [GHR94] D. M. Gabbay, I. M. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

- [GKV97] 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.
- [GMS03a] 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.
- [GMS03b] V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.
- [GMS04] 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.
- [GMSS06] 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.
- [GO07] 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.
- [GPO06a] J. Golińska-Pilarek and E. Orłowska. Relational logics and their applications. submitted, 2006.
- [GPO06b] J. Golińska-Pilarek and E. Orłowska. Tableaux and Dual Tableaux: Transformation of proofs. *Studia Logica*, 2006. to appear.
- [HMM83] J. Y. Halpern, Z. Manna, and B. C. Moszkowski. A Hardware Semantics Based on Temporal Intervals. In *Proc. of the 10th Colloquium on Automata, Languages and Programming*, volume 154 of *LNCS*, pages 278–291, 1983.
- [HS91] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
- [IK89] N. Immerman and D. Kozen. Definability with Bounded Number of Bound Variables. *Information and Computation*, 83(2):121–139, 1989.
- [Kam79] H. Kamp. Events, Instants and Temporal Reference. In R. Bäuerle, U. Egli, and C. Schwarze, editors, *Semantics from different points of view*, pages 376–417. de Gruyter, 1979.
- [KMMP93] 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.

- [KMO98] B. Konikowska, C. Morgan, and E. Orłowska. A relational formalisation of arbitrary finite-valued logics. *Logic Journal of IGPL*, 6(5):755–774, 1998.
- [Lad77] R. E. Ladner. The Computational Complexity of Provability in Systems of Modal Propositional Logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.
- [LM87] P. B. Ladkin and R. Maddux. The algebra of convex time intervals. Technical Report KES.U.87.2, Kestrel Institute, Palo Alto, California, 1987.
- [Lod00] K. Lodaya. Sharpening the Undecidability of Interval Temporal Logic. In *Proc. of 6th Asian Computing Science Conference*, volume 1961 of *LNCS*, pages 290–298. Springer, 2000.
- [LP00] O. Lichtenstein and A. Pnueli. Propositional Temporal Logic: decidability and completeness. *Logic Journal of the IGPL*, 8(1):55–85, 2000.
- [Mac97] W. MacCaull. Relational proof theory for linear and other substructural logics. *Logic Journal of IGPL*, 5:673–697, 1997.
- [Mac98] W. MacCaull. Relational semantics and a relational proof system for full Lambek Calculus. *Journal of Symbolic Logic*, 63(2):623–637, 1998.
- [MO02] W. MacCaull and E. Orłowska. Correspondence results for relational proof systems with applications to the Lambek Calculus. *Studia Logica*, 71:279–304, 2002.
- [Mon05] 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.
- [Mor75] M. Mortimer. On languages with two variables. *Zeitschr. f. math. Logik u. Grundlagen d. Math.*, 21:135–140, 1975.
- [Mos83] B. C. Moszkowski. *Reasoning about digital circuits*. Tech. rep. stan-cs-83-970, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.
- [MSV02] 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.
- [Orł88] E. Orłowska. Relational interpretation of modal logics. In H. Andreka, D. Monk, and I. Nemeti, editors, *Algebraic Logic*, volume 54 of *Colloquia Mathematica Societatis Janos Bolyai*, pages 443–471, North Holland, Amsterdam, 1988.

- [Orł94] E. Orłowska. Relational semantics for non-classical logics: Formulas are relations. In J. Woleński, editor, *Philosophical Logic in Poland*, pages 167–186. Kluwer, 1994.
- [Orł95] E. Orłowska. Temporal logics in a relational framework. In L. Bolc and A. Szalas, editors, *Time and Logic – a Computational Approach*, pages 249–277. University College London Press, 1995.
- [Orł96] E. Orłowska. Relational proof systems for modal logics. In H. Wansing, editor, *Proof Theory of Modal Logics*, pages 55–77. Kluwer, 1996.
- [Ott01] M. Otto. Two variable first-order logic over ordered domains. *Journal of Symbolic Logic*, 66(2):685–702, 2001.
- [Pae89] B. Paech. Gentzen-Systems for Propositional Temporal Logics. In *Proceedings of CSL '88, 2nd Workshop on Computer Science Logic*, volume 385 of *LNCS*, pages 240–253. Springer, 1989.
- [Pnu77] A. Pnueli. The Temporal Logic of Programs. In *Proc. of the 18th Annual IEEE Symposium on Foundations of Computer Science*, pages 46–57, 1977.
- [PPH98] J. Penix, C. Pecheur, and K. Havelund. Using model checking to validate AI planner domain models. In *Proc. of the 23rd Annual Software Engineering Workshop*, NASA Goddard, 1998.
- [PS81] A. Pnueli and R. Sherman. Semantic tableau for temporal logic. Technical Report CS81–82, The Weizmann Institute, 1981.
- [Rau83] W. Rautenberg. Modal Tableau Calculi and Interpolation. *Journal of Philosophical Logics*, 12:403–423, 1983.
- [RS63] H. Rasiowa. and R. Sikorski. *The Mathematics of Metamathematics*. Polish Scientific Publishers, Warsaw, 1963.
- [SC85] A. P. Sistla and E. M. Clarke. The Complexity of Propositional Linear Temporal Logics. *Journal of the ACM*, 32(3):733–749, 1985.
- [Sco62] D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:377–546, 1962.
- [Sha05] I. Shapirowsky. On PSPACE-decidability in Transitive Modal Logic. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic*, volume 5, pages 269–287. King’s College Publications, London, 2005.
- [SS03] I. Shapirowsky and V. Shehtman. Chronological future modality in Minkowski spacetime. In P. Balbiani, N. Y. Suzuki, F. Wolter, and M. Zakharyashev, editors, *Advances in Modal Logic*, volume 4, pages 437–459. King’s College Publications, London, 2003.

-
- [vB91] J. van Benthem. *The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse, Second Edition*. Kluwer, 1991.
- [Ven90] Y. Venema. Expressiveness and Completeness of an Interval Tense Logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990.
- [Ven91] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.
- [Vit05] N. Vitacolonna. *Intervals: Logics, Algorithms and Games*. PhD thesis, Department of Mathematics and Computer Science, University of Udine, Italy, 2005. PhD Thesis Series CS 2005/4.
- [Wol85] P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 28:119–136, 1985.