Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 231 (2009) 131–151
www.elsevier.com/locate/entcs

Complete and Terminating Tableau for the Logic of Proper Subinterval Structures Over Dense Orderings

Davide Bresolin[1]

Department of Computer Science, University of Verona, Verona, Italy

Valentin Goranko[2]

School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa

Angelo Montanari [3]

Department of Mathematics and Computer Science, University of Udine, Udine, Italy

Pietro Sala [4]

Department of Mathematics and Computer Science, University of Udine, Udine, Italy

Abstract

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

Keywords: Interval Temporal Logic, Tableau Method, Dense Structure.

[1] Email: davide.bresolin@univr.it
[2] Email: goranko@maths.wits.ac.za
[3] Email: angelo.montanari@dimi.uniud.it
[4] Email: pietro.sala@dimi.uniud.it

doi:10.1016/j.entcs.2009.02.033

1 Introduction

In interval temporal logics undecidability is usually the case (see, for instance, [12,14]), while decidability is a rare exception. The quest for decidable fragments and systems of temporal logics with interval-based semantics is one of the main research problems in the area of interval logics. Several decidability results have been established previously by reduction to point-based logics, either by way of direct translation or by restriction of the semantics, e.g., imposing locality, homogeneity, or other principles that essentially reduce it to point-based semantics [1,2,3,10,11,13,15].
Only recently some decidability results of genuinely interval-based logics have been established [4,5,6,7,8,9]. In particular, in [4] we have developed a sound, complete and terminating tableau for the logic $\mathrm{D}_{\dot\sqsubset}$ of *strict subintervals* (with both endpoints strictly inside the current interval) over dense linear orderings, by defining a class of pseudo-models and proving finite model property with respect to such pseudo-models.

Here we consider the interval logic $\mathrm{D}_{\sqsubset}$ of *proper subintervals*, that is, subintervals different from the current interval, over dense linear orderings and we develop a similar technique to devise a tableau-based decision procedure for that logic. Despite the strong similarity with our previous work, the case of proper subintervals turned out to be essentially more complicated. The presence of the special families of beginning subintervals and ending subintervals of a given interval in a structure with proper subinterval relation causes substantial distinction of the semantics from the case of interval structures with strict subinterval relation studied in [4], further leading to considerable complications in the constructions of both pseudo-models and tableaux. For instance, the formula $(\langle D\rangle p \wedge \langle D\rangle q) \rightarrow \langle D\rangle(\langle D\rangle p \wedge \langle D\rangle q)$ is valid in $\mathrm{D}_{\dot\sqsubset}$ but not in $\mathrm{D}_{\sqsubset}$ (for, p and q may only be satisfied in respectively beginning and ending subintervals). Furthermore, the formula

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

can only be satisfied in a $\mathrm{D}_{\sqsubset}$-structure, as it forces p to be true at some beginning and at some ending subintervals, a requirement which cannot be imposed in $\mathrm{D}_{\dot\sqsubset}$. Note, however, that while $\mathrm{D}_{\sqsubset}$ can refer to beginning or ending intervals, *it cannot differentiate between these*. This is a subtle but crucial detail: as shown by Lodaya [14], the interval logic BE with modalities respectively for beginning and ending subintervals is *undecidable over the class of dense orderings*.

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

2 Structures for $D_\sqsubset$ formulas

2.1 Syntax and semantics of $D_\sqsubset$

Let $\mathbb{D} = \langle D, < \rangle$ be a dense linear order. An *interval* over $\mathbb{D}$ is an ordered pair $[b, e]$, where $b < e$. We denote the set of all intervals over $\mathbb{D}$ by $\mathbb{I}(\mathbb{D})$. We consider the *proper* (i.e., irreflexive) *subinterval relation*, denoted by $\sqsubset$, defined as follows: $[d_k, d_l] \sqsubset [d_i, d_j]$ if and only if $d_i \leq d_k$, $d_l \leq d_j$ and $[d_k, d_l] \neq [d_i, d_j]$. We shall write $[d_k, d_l] \sqsubseteq [d_i, d_j]$ as a shorthand for $[d_k, d_l] \sqsubset [d_i, d_j] \vee [d_k, d_l] = [d_i, d_j]$.

The language of the modal logic $D_\sqsubset$ of interval structures with proper subinterval relation consists of a set $\mathcal{AP}$ of propositional letters, the propositional connectives $\neg$ and $\vee$, and the modal operator $\langle D \rangle$. The other propositional connectives, as well as the logical constants $\top$ (*true*) and $\bot$ (*false*) and the dual modal operator $[D]$, are defined as usual. Formulas of $D_\sqsubset$ are defined as follows: $\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle D \rangle \varphi$. The semantics of $D_\sqsubset$ is based on *interval models* $\mathbf{M} = \langle \mathbb{I}(\mathbb{D}), \sqsubset, \mathcal{V} \rangle$. The *valuation function* $\mathcal{V} : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})}$ assigns to every propositional variable p the set of intervals $\mathcal{V}(p)$ over which p holds. The semantics of $D_\sqsubset$ is recursively defined by the satisfiability relation $\Vdash$ as follows:

- for every propositional variable $p \in \mathcal{AP}$, $\mathbf{M}, [d_i, d_j] \Vdash p$ iff $[d_i, d_j] \in \mathcal{V}(p)$;
- $\mathbf{M}, [d_i, d_j] \Vdash \neg\psi$ iff $\mathbf{M}, [d_i, d_j] \not\Vdash \psi$;
- $\mathbf{M}, [d_i, d_j] \Vdash \psi_1 \vee \psi_2$ iff $\mathbf{M}, [d_i, d_j] \Vdash \psi_1$ or $\mathbf{M}, [d_i, d_j] \Vdash \psi_2$;
- $\mathbf{M}, [d_i, d_j] \Vdash \langle D \rangle \psi$ iff there exists $[d_k, d_l] \in \mathbb{I}(\mathbb{D})$ such that $[d_k, d_l] \sqsubset [d_i, d_j]$ and $\mathbf{M}, [d_k, d_l] \Vdash \psi$.

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

2.2 Fulfilling $D_\sqsubset$-structures

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

Definition 2.1 Given a $D_\sqsubset$-formula φ, a φ*-atom* is a subset A of $\mathrm{CL}(\varphi)$ such that:
(i) for every $\psi \in \mathrm{CL}(\varphi)$, $\psi \in A$ if and only if $\neg\psi \notin A$, and
(ii) for every $\psi_1 \vee \psi_2 \in \mathrm{CL}(\varphi)$, $\psi_1 \vee \psi_2 \in A$ if and only if $\psi_1 \in A$ or $\psi_2 \in A$.

Definition 2.2 Given a $D_\sqsubset$-formula φ and a φ-atom $A \in \mathcal{A}_\varphi$, the *set* $\mathrm{REQ}(A)$ *of (temporal) requests of* A is the set $\{\langle D \rangle \psi \in \mathrm{CL}(\varphi) : \langle D \rangle \psi \in A\}$.

We denote the set of all φ-atoms by A_φ and the set of all $\langle D \rangle$-formulas in $CL(\varphi)$ by REQ_φ. Then, we define the binary relation $D_\varphi \subseteq A_\varphi \times A_\varphi$, such that $A \; D_\varphi \; A'$ if and only if for every $[D]\psi$ in $CL(\varphi)$, if $[D]\psi \in A$, then $\psi \in A'$.

Given an interval $[b, e]$, a *beginning subinterval* of $[b, e]$ is an interval $[b, e']$, with $e' < e$, an *ending subinterval* of $[b, e]$ is an interval $[b', e]$, with $b < b'$, and an *internal subinterval* of $[b, e]$ is an interval $[b', e']$, with $b < b'$ and $e' < e$. To represent infinite chains of beginning (resp., ending) subintervals of a given interval, we need to

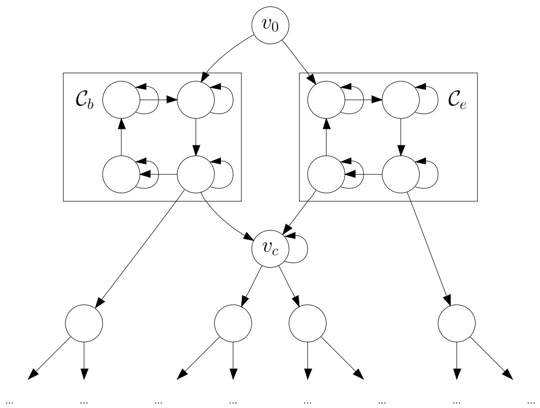

Figure 1. An example of $D_\sqsubset$-graph.

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

Definition 2.3 A finite directed graph $\mathbb{G} = \langle V, E\rangle$ is a $D_\sqsubset$-*graph* if:

(i) there exists an irreflexive vertex $v_0 \in V$, called the *root* of $\mathbb{G}$, such that any other vertex $v \in V$ is reachable from it;

(ii) every irreflexive vertex $v \in V$ has exactly two clusters as successors: a *beginning successor cluster* $\mathcal{C}_b$ and an *ending successor cluster* $\mathcal{C}_e$;

(iii) $\mathcal{C}_b$ and $\mathcal{C}_e$ have a unique common successor v_c, which is a reflexive vertex;

(iv) every successor of v_c, different from v_c itself, is irreflexive;

(v) there exists at most one edge exiting the clusters $\mathcal{C}_b$ and $\mathcal{C}_e$ and reaching an irreflexive node;

(vi) apart from the edge leading to v_c, there are no edges exiting from $\mathcal{C}_b$ (resp. $\mathcal{C}_e$) that reach a reflexive vertex.

Figure 1 depicts a portion of a $D_\sqsubset$-graph. The root v_0 has two successor clusters $\mathcal{C}_b$ and $\mathcal{C}_e$ of four vertices each. Both $\mathcal{C}_b$ and $\mathcal{C}_e$ have exactly one irreflexive successor. Their common reflexive successor v_c has two irreflexive successors.

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

Definition 2.4 A $D_\sqsubset$-*structure* is a quadruple $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$, where:

(i) $\langle V, E\rangle$ is a $D_\sqsubset$-graph;

(ii) $\mathcal{L} : V \to \mathcal{A}_\varphi$ is a *labeling function* that assigns to every vertex $v \in V$ an atom $\mathcal{L}(v)$ such that for every edge $(v, v') \in E$, $\mathcal{L}(v)\ D_\varphi\ \mathcal{L}(v')$;

(iii) $\mathcal{B} : V \to 2^{\mathrm{REQ}_\varphi}$ and $\mathcal{E} : V \to 2^{\mathrm{REQ}_\varphi}$ are mappings that assign to every vertex the sets of its *beginning* and *ending requests*, respectively;

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

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

Beginning and ending requests associated with a vertex v can be viewed as requests that must be satisfied over respectively beginning and ending subintervals of any interval corresponding to v (possibly over both of them), but not over its internal subintervals.

Every $\mathrm{D}_{\sqsubset}$-structure can be regarded as a Kripke model for $\mathrm{D}_{\sqsubset}$, where the valuation is determined by the labeling.

Definition 2.5 A $\mathrm{D}_{\sqsubset}$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ is *fulfilling* if for every $v \in V$ and every $\langle D\rangle\psi \in \mathcal{L}(v)$, there exists $v' \in V$ such that v' is a descendant of v and $\psi \in \mathcal{L}(v')$.

Theorem 2.6 *Let φ be a $\mathrm{D}_{\sqsubset}$-formula which is satisfied in an interval model. Then, there exists a fulfilling $\mathrm{D}_{\sqsubset}$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ for φ.*

Proof Let $\mathbf{M} = \langle \mathbb{I}(\mathbb{D}), \sqsubset, \mathcal{V}\rangle$ be an interval model and let $[b_0, e_0] \in \mathbb{I}(\mathbb{D})$ be an interval such that $\mathbf{M}, [b_0, e_0] \Vdash \varphi$. We recursively build a fulfilling $\mathrm{D}_{\sqsubset}$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ for φ as follows.

We start with the one-node graph $\langle\{v_0\}, \emptyset\rangle$ and a labeling function $\mathcal{L}$ such that $\mathcal{L}(v_0) = \{\psi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_0, e_0] \Vdash \psi\}$. Then, we partition the set $\mathrm{REQ}(\mathcal{L}(v_0))$ into the following three sets of formulas:

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

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

Internal requests: $I_{v_0} = (\mathrm{REQ}(\mathcal{L}(v_0)) \setminus B_{v_0}) \setminus E_{v_0}$, that is, the set of all $\langle D\rangle\xi \in \mathrm{REQ}(\mathcal{L}(v_0))$ such that ξ is satisfied over internal subintervals of $[b_0, e_0]$.

We put $\mathcal{B}(v_0) = B_{v_0}$ and $\mathcal{E}(v_0) = E_{v_0}$. Then, for every formula $\langle D\rangle\psi \in \mathcal{L}(v_0)$, we choose an interval $[b_\psi, e_\psi]$, with $[b_\psi, e_\psi] \sqsubset [b_0, e_0]$, such that $\mathbf{M}, [b_\psi, e_\psi] \Vdash \psi$. If

$\langle D\rangle\psi \in I_{v_0}$, then $b_0 < b_\psi < e_\psi < e_0$, else if $\langle D\rangle\psi \in B_{v_0}$, then $b_0 = b_\psi < e_\psi < e_0$, otherwise ($\langle D\rangle\psi \in E_{v_0}$) $b_0 < b_\psi < e_\psi = e_0$.

Since $\mathbb{D}$ is a dense ordering and $\mathrm{CL}(\varphi)$ is a finite set of formulas, there exist two beginning intervals $[b_0, e_1]$ and $[b_0, e_2]$ such that:

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

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

The ending successor cluster $\mathcal{C}_e$ of v_0 is built in the very same way.

To complete the first phase of the construction, we must introduce the common reflexive successor v_c of $\mathcal{C}_b$ and $\mathcal{C}_e$. Since $\mathbb{D}$ is a dense ordering and $\mathrm{CL}(\varphi)$ is a finite set of formulas, there exist two intervals $[b_3, e_3]$ and $[b_4, e_4]$ such that:

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

We add a new vertex v_c, together with the edges (v_b, v_c), (v_e, v_c), and (v_c, v_c), and we put $\mathcal{L}(v_c) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_3, e_3] \Vdash \xi\}$, $\mathcal{B}(v_c) = \mathcal{E}(v_c) = \emptyset$.

For every formula $\langle D\rangle\psi \in I_{v_0}$, we add a new vertex v_ψ and an edge (v_c, v_ψ), and we define its labeling as $\mathcal{L}(v_\psi) = \{\xi \in \mathrm{CL}(\varphi) : \mathbf{M}, [b_\psi, e_\psi] \Vdash \xi\}$.

Then, we recursively apply the above procedure to the irreflexive vertices we have introduced. To keep the construction finite, whenever there exists an irreflexive vertex $v' \in V$ such that $\mathcal{L}(v_\psi) = \mathcal{L}(v')$ for some v_ψ, we simply add an edge to v' instead of creating a new vertex v_ψ and an edge entering it. Since the set of atoms is finite, the construction is guaranteed to terminate. □

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

Definition 2.7 Let $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ be a $\mathrm{D}_\sqsubset$-structure. The function $f_\mathbf{S} : \mathbb{I}([0, 1]) \mapsto V$ is defined recursively as follows. First, $f_\mathbf{S}([0, 1]) = v_0$. Now, let $[b, e]$

be an interval such that $f_{\mathbf{S}}([b, e]) = v$ and $f_{\mathbf{S}}$ has not been yet defined over any of its subinterval. We distinguish two cases.

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

(i) we put $f_{\mathbf{S}}([b, b+p]) = v_b^{max}$ and $f_{\mathbf{S}}([e-p, e]) = v_e^{max}$;

(ii) for every $i = 1, \ldots, k$, we put $f_{\mathbf{S}}([b + 2ip, b + (2i+1)p]) = v_i$;

(iii) for every $i = 1, \ldots, k+1$, we put $f_{\mathbf{S}}([b + (2i-1)p, b + 2ip]) = v_c$;

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

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

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

(i) *v_c has no successors apart from itself.* In such a case, we put $f_{\mathbf{S}}([b', e']) = v_c$ for every subinterval $[b', e']$ of $[b, e]$.

(ii) *v_c has at least one successor different from itself.* Let $v_c^1, \ldots, v_c^k$ be the k successors of v_c different from v_c. We consider the intervals defined by the points $b, b+p, b+2p \ldots, b+2kp, b+(2k+1)p = e$, with $p = \frac{e-b}{2k+1}$. The function $f_{\mathbf{S}}$ over such intervals is defined as follows:

- for every $i = 1, \ldots, k$, we put $f_{\mathbf{S}}([b + (2i-1)p, b + 2ip]) = v_c^i$.
- for every $i = 0, \ldots, k$, we put $f_{\mathbf{S}}([b + 2ip, b + (2i+1)p]) = v_c$.

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

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

Lemma 2.8

(i) *For every interval $[b, e] \in \mathbb{I}([0, 1])$, if $f_{\mathbf{S}}([b, e]) = v$ and v' is reachable from v, then there exists an interval $[b', e']$ such that $f_{\mathbf{S}}([b', e']) = v'$ and $[b', e'] \sqsubset [b, e]$.*

(ii) *For every pair of intervals $[b, e]$ and $[b', e']$ in $\mathbb{I}([0, 1])$ such that $[b', e'] \sqsubset [b, e]$, we have that for every formula $[D]\psi \in \mathcal{L}(f_{\mathbf{S}}([b, e]))$, both ψ and $[D]\psi$ belong to*

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

$\mathcal{L}(f_{\mathbf{S}}([b', e']))$.

Proof Condition 1 can be easily proved by observing that it trivially holds for all successors of v by definition of $f_{\mathbf{S}}$ and then extending the result to every descendant v' of v by induction on the length of the shortest path from v to v'.

As for condition 2, let $[b, e]$ and $[b', e']$ be two intervals in $\mathbb{I}([0, 1])$ such that $[b', e'] \sqsubset [b, e]$, $v = f_{\mathbf{S}}([b, e])$, and $v' = f_{\mathbf{S}}([b', e'])$. If v' is a descendant of v in the $\mathrm{D}_{\sqsubset}$-graph, then condition 2 holds by definition of D_{φ}. When we apply the construction step defined by Case 1, Point 4, of Definition 2.7, it may happen that $[b', e'] \sqsubset [b, e]$ but v' is not reachable from v in the $\mathrm{D}_{\sqsubset}$-graph. In such a case, both $[b, e]$ and $[b', e']$ are internal subintervals, and thus, by definition of the labeling functions $\mathcal{B}$ and $\mathcal{E}$, condition 2 is satisfied. □

Theorem 2.9 *Given a fulfilling* $\mathrm{D}_{\sqsubset}$*-structure* $\mathbf{S}$ *for* φ*, there exists an interval model* $\mathbf{M}_{\mathbf{S}} = \langle \mathbb{I}([0, 1]), \sqsubset, \mathcal{V} \rangle$ *over the rational interval* $[0, 1]$ *such that* $\mathbf{M}_{\mathbf{S}}, [0, 1] \Vdash \varphi$.

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

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

The atomic case immediately follows from definition of $\mathcal{V}$; the Boolean cases follow from the definition of atom; finally, the case of temporal formulas follows from Lemma 2.8. This allows us to conclude that $\mathbf{M}_{\mathbf{S}}, [0, 1] \models \varphi$. □

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

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

Theorem 2.10 *For every satisfiable* $\mathrm{D}_{\sqsubset}$*-formula* φ*, there exists a fulfilling* $\mathrm{D}_{\sqsubset}$*-structure with breadth and depth bounded by* $2 \cdot |\varphi|$.

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

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

The execution of the first removal process produces a $\mathrm{D}_{\sqsubset}$-structure where the size of every cluster is at most $|\varphi|$ and every vertex in a cluster of size at least 2 fulfills some ψ formulas which are not fulfilled elsewhere, while the execution of the second removal process produces a $D_{\sqsubset}$-structure where every vertex has at most $|\varphi|$ immediate successors.

Since whenever we exit from a cluster or we move from a reflexive node to an irreflexive one the number of requests strictly decreases, we can conclude that the length of every loop-free path is at most $2 \cdot |\varphi|$. $\square$

As a direct consequence of Theorem 2.10, we have that a fulfilling $D_{\sqsubset}$-structure for a formula φ (if any) can be generated and explored by a non-deterministic procedure that uses only a polynomial amount of space. This gives the following complexity bound to the decision problem for $D_{\sqsubset}$.

Theorem 2.11 *The decision problem for* $D_{\sqsubset}$ *is in* $PSPACE$.

The very same reduction that has been used to prove $D_{\sqsubset\!\!\cdot}$ PSPACE hardness in [4] can be applied to $D_{\sqsubset}$, thus proving the PSPACE completeness of the satisfiability problem for $D_{\sqsubset}$.

3 The tableau method for $D_{\sqsubset}$

In this section we present a tableau system for $D_{\sqsubset}$. From the model-theoretic results in the previous section, we have that a $D_{\sqsubset}$-formula φ is satisfiable if and only if there exists a fulfilling $D_{\sqsubset}$-structure for it. The tableau method attempts systematically to build such a structure if there is any, returning "satisfiable" if it succeeds and "unsatisfiable" otherwise.

The nodes of the tableau are sets of locally consistent formulas (i.e., parts of atoms). At the root of the tableau, we place a set containing only the formula φ the satisfiability of which is being tested. We then proceed recursively to expand the tableau, following the expansion rules described below. Every disjunctive branch of the tableau describes an attempt to construct a fulfilling $D_{\sqsubset}$-structure for the atom at the root. Going down the branch roughly corresponds to going deeper into subintervals of the interval corresponding to the root. The applicability of an expansion rule at a given node depends on the formulas in the node and on the part of $D_{\sqsubset}$-structure we are building. The expansion of the tableau proceeds as follows.

(i) We start with the *current vertex* (at the beginning, the root) v_0 of the $D_{\sqsubset}$-structure that is being constructed and we apply the usual Boolean rules to decompose Boolean operators.

(ii) Then, we impose a suitable marking on $\langle D \rangle$-formulas to partition them into four sets: the set of formulas that are satisfied only on beginning subintervals, that of formulas that are satisfied only on ending subintervals, that of formulas that are satisfied both on beginning and ending subintervals, and that of formulas that are satisfied on internal subintervals.

(iii) The third phase of the procedure is the construction of the first vertex v_b of the beginning successor cluster $\mathcal{C}_b$, the first vertex v_e of the ending successor cluster $\mathcal{C}_e$, and their common successor v_c.

(iv) Next, we proceed in parallel with the construction of the clusters $\mathcal{C}_b$ and $\mathcal{C}_e$ by guessing the $\langle D\rangle$-formulas from the set $REQ(\mathcal{L}(v_0))$ that should be satisfied inside each of them.

(v) Then, we build the irreflexive successor v_b^{max} of $\mathcal{C}_b$, the irreflexive successor v_e^{max} of $\mathcal{C}_e$, and the irreflexive successors of v_c, if needed, and proceed recursively with their expansion from Step 1 above.

During the expansion of the tableau, we restrict our search to models with the property stated in Theorem 2.10. In particular, during the construction of a cluster we explicitly satisfy only those $\langle D\rangle$-formulas that should be satisfied inside the cluster and can never be satisfied outside it. In this way we have the following advantages:

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

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

3.1 The rules of the tableau.

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

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

This notation has the following intuitive meaning. The markings $\langle D\rangle^M\psi$, $\langle D\rangle^B\psi$, $\langle D\rangle^E\psi$, and $\langle D\rangle^{BE}$ appear when we try to construct an irreflexive interval node and we guess that the formula $\langle D\rangle\psi$ should be satisfied over an internal (middle) subinterval, only over a beginning subinterval, only over an ending subinterval, or both over a beginning and over an ending (but not over middle) subinterval of the current one. The markings $\langle D\rangle^{BC}\psi$ or $\langle D\rangle^{BNC}\psi$ (resp. $\langle D\rangle^{EC}\psi$,$\langle D\rangle^{ENC}\psi$) substitute a previously marked $\langle D\rangle^B\psi$ (resp. $\langle D\rangle^E\psi$) formula when we try to construct a beginning cluster and we guess that the formula ψ should be satisfied in the current cluster ($\langle D\rangle^{BC}\psi$ marking) or not ($\langle D\rangle^{BNC}\psi$ marking). The marking is only used for bookkeeping purposes, to facilitate the correct choice of the rules to be applied. It does not affect the existence of a contradiction; we say that a *node is closed* iff once we remove the marking from every formula in it, it then contains both ψ and $\neg\psi$ for some $\psi \in CL(\varphi)$.

Given a set Φ of possibly marked formulas, the set $TF(\Phi)$ (the *temporal fragment of* Φ) is the set of all the formulas in Φ of the types $\langle D\rangle\psi$ and $[D]\psi$ (ignoring the markings). Given a set of formulas Γ, we use $(D)\Gamma$, where $(D) \in \{[D], \langle D\rangle, \langle D\rangle^M, \langle D\rangle^B, \langle D\rangle^{BC}, \langle D\rangle^{BNC}, \langle D\rangle^E, \langle D\rangle^{EC}, \langle D\rangle^{ENC}, \langle D\rangle^{BE}\}$, as a shorthand for $\{(D)\psi \mid \psi \in \Gamma\}$. Likewise, $\neg\Gamma$ stands for $\{\neg\psi \mid \psi \in \Gamma\}$ and $\Gamma \vee (D)\Gamma$ for $\{\psi \vee (D)\psi \mid \psi \in \Gamma\}$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Then, we apply again all steps from the beginning, with only a little difference in the application of the Marking Rule. The Completion Rule may produce some requests $\langle D\rangle\psi$ devoid of any markings. For all these requests, we must check whether they have been marked as $\langle D\rangle^{B}\psi$ or $\langle D\rangle^{E}\psi$ in an ancestor of the current node and, if this is the case, we must guarantee the downward propagation of their markings. To this end, before applying the Marking Rule, we apply the following **Persistent Beginning** and **Persistent Ending Rules**:

$$\frac{\Phi, \langle D\rangle\psi}{\Phi, \langle D\rangle^{B}\psi} \qquad \frac{\Phi, \langle D\rangle\psi}{\Phi, \langle D\rangle^{E}\psi}$$

whenever $\langle D\rangle^{B}\psi$ (resp., $\langle D\rangle^{E}\psi$) belongs to an ancestor of the current node.

3.2 Building the tableaux.

A tableau for a $\mathrm{D}_{\sqsubset}$-formula φ is a finite graph $\mathcal{T} = \langle V, E\rangle$, whose vertices are subsets of $\mathrm{CL}(\varphi)$ and whose edges are generated by the application of expansion rules. The construction of the tableau starts with the *initial tableau*, which is the single node graph $\langle\{\{\varphi\}\}, \emptyset\rangle$. To describe such a construction process, we take advantage of macronodes, which can be viewed as the counterpart of vertices of $\mathrm{D}_{\sqsubset}$-structures.

Given a set $V' \subseteq V$, let $E(V')$ be the restriction of E to vertices in V. Moreover, let the Reflexive Step, Middle Step, Build Beginning/Ending Cluster and Exit Beginning/Ending Cluster rules be called **Step Rules**. Macronodes are defined as follows.

Definition 3.1 Let $\langle V, E\rangle$ be a tableau for a $\mathrm{D}_{\sqsubset}$-formula φ. A *macronode* is a set $V' \subseteq V$ such that:

- $\langle V', E(V')\rangle$ is a tree;
- the root of $\langle V', E(V')\rangle$ is either the initial node of the tableau or a node generated by an application of a Step Rule;
- every edge in $E(V')$ is generated by the application of an expansion rule which is not a Step Rule;
- the only expansion rule that can be applied to the leaves of $\langle V', E(V')\rangle$ is a Step Rule.

A macronode m is *reflexive* if its root is generated by the application of the Reflexive Step Rule or of the Build Beginning/Ending Cluster Rules; otherwise, it is *irreflexive*.

We say that a rule is applicable to a node n if it generates at least one successor node. The construction of a tableau for a $\mathrm{D}_{\sqsubset}$-formula φ starts with the initial tableau $\langle\{\{\varphi\}\}, \emptyset\rangle$ and proceeds by applying the following *expansion strategy* to the leaves of the current tableau, until it cannot be applied anymore.

Apply the first rule in the list whose condition is satisfied:

(i) a Boolean Rule is applicable;

(ii) the Completion Rule is applicable;

(iii) the node belongs to an irreflexive macronode and the Persistent Beginning Rule is applicable;

(iv) the node belongs to an irreflexive macronode and the Persistent Ending Rule is applicable;

(v) the node belongs to an irreflexive macronode and the Marking Rule is applicable;

(vi) the node belongs to an irreflexive macronode and the Reflexive Step Rule is applicable;

(vii) the node belongs to a reflexive macronode with only M markings and the Middle Step Rule is applicable;

(viii) the node belongs to a reflexive macronode with B markings or E markings and the Build Beginning/Ending Cluster Rules are applicable;

(ix) the node belongs to a reflexive macronode with B markings or E markings and the Exit Beginning/Ending Cluster Rules are applicable.

Termination is ensured by the following *looping conditions*:

- if an application of the Reflexive Rule generates a node which is the root of an existing reflexive macronode, then add an edge from the current node to this node instead of creating the new one.
- if the Middle Step Rule is applied to a node n and one of the successor nodes it generates, say n', is such that $TF(n') = TF(n)$, then add the edge (n', n) to the tableau. Do not apply any expansion rule to n'.

We say that a node n in a tableau is *closed* if one of the following conditions holds:

- there exists ψ such that both ψ and $\neg\psi$ belong to n;
- a Middle Step Rule or a Reflexive Step Rule have been applied to n and *at least one* of its successors is closed;
- a rule different from the Middle Step Rule and the Reflexive Step Rule has been applied to n and *all* its successors are closed;
- n is a descendant of a node n' to which an Exit Beginning/Ending Cluster Rule has been applied and $TF(n') = TF(n)$.

A node in a tableau is *open* if it is not closed. A tableau is *open* if and only if its root is open. We will prove that a formula is satisfiable if and only if there exists an open tableau for it.

As for computational complexity, it is not difficult to show that the proof of Theorem 2.10 can be adapted to the proposed tableau method. The only difference is that at any step of the tableau construction we either expand a node or mark one of its formulas. As a consequence, any node of a $D_{\sqsubset}$-structure corresponds to a path of at most $|\varphi|$ nodes in the tableau. Hence, the depth of the tableau is bounded by $2 \cdot |\varphi|^2$. Since the breadth of the tableau is $2 \cdot |\varphi|$, we can conclude that the proposed tableau-based decision procedure is in $PSPACE$ (and thus optimal).

Theorem 3.2 *(Complexity) The proposed tableau procedure is in $PSPACE$.*

3.3 Example of application.

Here we give an example of the above-described expansion strategy at work. Consider the formula $\varphi = \langle D\rangle p \wedge \langle D\rangle q \wedge [D]\neg(\langle D\rangle p \wedge \langle D\rangle q)$, which states that the given interval has a subinterval where p holds and a subinterval where q holds, but no subintervals covering both of them. It is easy to see that in any model for this formula p and q respectively hold in a beginning and an ending subinterval only, or vice versa. Part of the tableau for φ is depicted in Figure 2. Due to space limitations, we restrict our attention to the non-closed region of the tableau and we skip the details about the application of Boolean Rules. We start with the root A, whose temporal fragment is complete, and we apply the Marking Rule. For the sake of conciseness, we only consider a correct marking, which inserts $\langle D\rangle^B p$ and $\langle D\rangle^E q$ in B. Once all $\langle D\rangle$-formulas have been marked, we apply the Reflexive Step Rule, that generates the three successors of B. The first successor is node C that contains the request $\langle D\rangle^B p$ and the negation of the request $\langle D\rangle^E q$, namely, $[D]\neg q$. The second one is node E that contains the request $\langle D\rangle^E q$ and the negation of the request $\langle D\rangle^B p$, namely, $[D]\neg p$. The third one is node D that contains the negation of the two requests (such a node represents the middle reflexive vertex of the corresponding $D_{\sqsubset}$-structure). Node D contains no $\langle D\rangle$-formulas and thus it cannot be expanded anymore. Since it does not include any contradiction, we declare it open. Consider now node C. According to the expansion strategy, we apply the Build Beginning Cluster Rule to $\langle D\rangle^B p$ in node C, that generates nodes F and G. Node F includes p and, accordingly, replaces $\langle D\rangle^B p$ with $\langle D\rangle^{BC} p$. It does not contain $\langle D\rangle^{BNC}$ formulas and no expansion rules are applicable to it. Since it does not include any contradiction, we declare it open. The same argument can be applied to nodes E and H. This allows us to conclude that the tableau is open (and thus φ is satisfiable).

To better explain the proposed tableau method, we include in Figure 2 additional nodes which are not strictly necessary to conclude that the tableau is open. This is the case with node G that replaces $\langle D\rangle^B p$ with $\langle D\rangle^{BNC} p$, thus postponing the satisfaction of p. According to the expansion strategy, we apply the Exit Beginning Cluster Rule to G, that generates the irreflexive node L. Such a node contains the formula $\langle D\rangle^B p \vee p$, stating that p is satisfied either in L or in some descendant of it. The application of the Or Rule to $\langle D\rangle^B p \vee p$ generates nodes M and N. Node M includes again the formula $\langle D\rangle^B p$ and, since $TF(M) = TF(G)$, we declare it closed. As for node N, that satisfies p, we apply the Completion Rule (neither $\langle D\rangle p$ nor $[D]\neg p$ belongs to N), that generates its two successors. The first successor turns

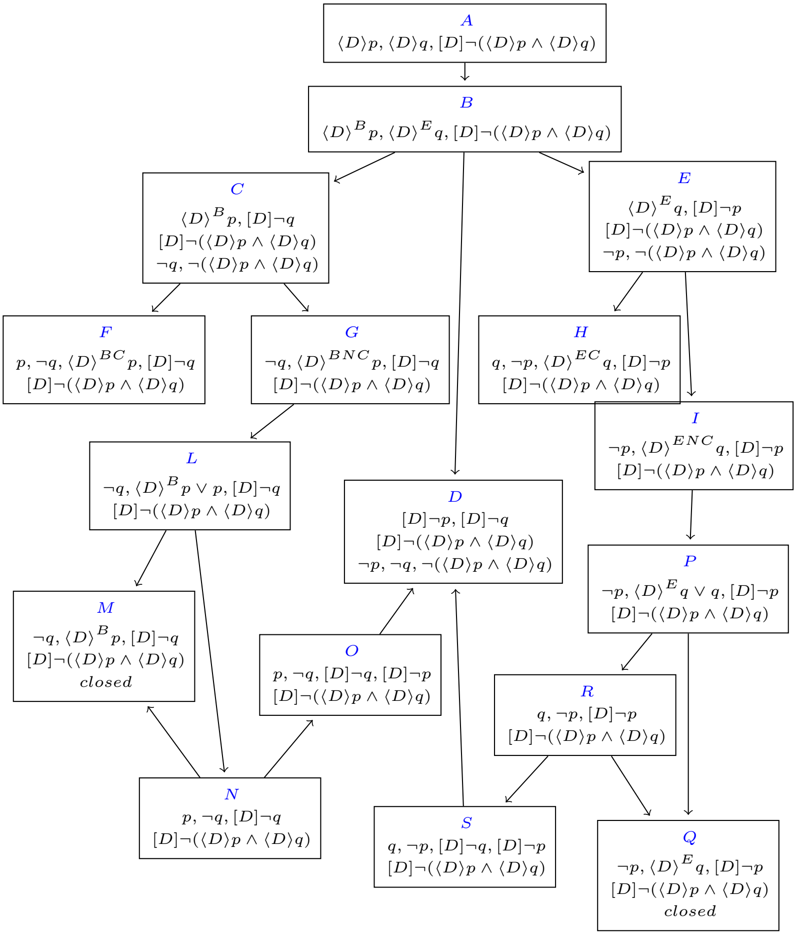

Figure 2. (Part of) the tableau for $\varphi = \langle D\rangle p \wedge \langle D\rangle q \wedge [D]\neg(\langle D\rangle p \wedge \langle D\rangle q)$.

out to be identical to M and thus we add an edge from N to M instead of adding a new node; the second successor is node O, with $TF(O) \subset TF(G)$. Then, we apply Reflexive Step Rule to node O. Since it does not contain any $\langle D\rangle$-formula, its three reflexive successors coincides with node D. Hence, we add an edge from O to D and we stop the expansion of (this part of) the tableau.

3.4 Soundness and completeness

We conclude the section by proving soundness and completeness of the tableau method.

Theorem 3.3 (SOUNDNESS) *Let φ be a $\mathrm{D}_{\sqsubset}$-formula and $\mathcal{T}$ be a tableau for it. If $\mathcal{T}$ is open, then φ is satisfiable.*

Proof We build a fulfilling $\mathrm{D}_{\sqsubset}$-structure $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ for φ step by step, starting from the root of $\mathcal{T}$ and proceeding according to the expansion rules that have been applied in the construction of the tableau.

Let n_0 be the root of $\mathcal{T}$. We generate the one-node $\mathrm{D}_{\sqsubset}$-graph $\langle\{v_0\}, \emptyset\rangle$ and we put formulas belonging to n_0 in $\mathcal{L}(v_0)$. Now, let n be an open node in $\mathcal{T}$ and let v be the corresponding vertex in the $\mathrm{D}_{\sqsubset}$-graph. The way in which we develop the $\mathrm{D}_{\sqsubset}$-structure depends on the expansion rule that has been applied to n during the construction of the tableau.

- *A Boolean Rule has been applied.* Then, at least one successor n' of n is open. We add formulas belonging to n' to $\mathcal{L}(v)$ and we proceed by taking into consideration the tableau node n' and the vertex v.
- *The Completion Rule has been applied.* Then, at least one successor n' of n is open. As in the previous case, we add formulas belonging to n' to $\mathcal{L}(v)$ and we proceed by taking into consideration the tableau node n' and the vertex v.
- *The Marking/Persistent Beginning/Persistent Ending Rule has been applied.* Let $\langle D\rangle\psi$ be the formula to which the rule has been applied and let n' be one of the open successors of n. Four cases may arise, depending on which marking has been applied to the considered formula in n':
  - if $\langle D\rangle^B\psi \in n'$, then we put $\langle D\rangle\psi \in \mathcal{B}(v)$;
  - if $\langle D\rangle^E\psi \in n'$, then we put $\langle D\rangle\psi \in \mathcal{E}(v)$;
  - if $\langle D\rangle^{BE}\psi \in n'$, then we add $\langle D\rangle\psi$ to both $\mathcal{B}(v)$ and $\mathcal{E}(v)$;
  - if $\langle D\rangle^M\psi \in n'$, then the marking does not influence the construction of the $\mathrm{D}_{\sqsubset}$-structure.

  In all cases, we proceed recursively by taking into consideration the tableau node n' and the current vertex v.
- *The Reflexive Step Rule has been applied.* Since $\mathcal{T}$ is open, all successors of n are open either. Let n_b, n_c, and n_e be the first, second, and third successor of n, respectively. We add three reflexive vertices v_b, v_c, and v_e to V and the edges (v, v_b), (v, v_e), (v_b, v_c), (v_e, v_c), (v_b, v_b), (v_c, v_c), and (v_e, v_e) to E. The labeling of v_b, v_c, and v_e is defined as follows: $\mathcal{L}(v_b) = n_b$, $\mathcal{L}(v_c) = n_c$, and $\mathcal{L}(v_e) = n_e$. We recursively apply the construction by taking into consideration the node n_b with the corresponding vertex v_b, the node n_c with the corresponding vertex v_c, and the node n_e with the corresponding vertex v_e.
- *The Middle Step Rule has been applied.* Since n is open, all its successors $n_1, ..., n_h$ are open either. We add h new vertices $v_1, ..., v_h$ to V and the edges $(v, v_1), ..., (v, v_h)$ to E, and we define their labeling in such a way that for $i = 1, \ldots, h$, $\mathcal{L}(v_i) = n_i$. We recursively apply the construction to every node n_i paired with the corresponding vertex v_i.
- *The Build Beginning/Ending Cluster Rule has been applied.* Suppose that the rule has been applied to a formula $\langle D\rangle^B\psi \in n$ (the case of $\langle D\rangle^E\psi$ is analogous) and let n' be an open successor of n. Two cases may arise:
  - (i) $\langle D\rangle^{BC}\psi \in n'$ ($\langle D\rangle\psi$ has been satisfied in the cluster). We introduce a new node v' in the cluster of v by adding the edges (v, v'), (v', v'), and (v', v) to E. The labeling $\mathcal{L}(v')$ of v' consists of the set of formulas belonging to n'. We proceed by taking into consideration the node n' and the corresponding vertex v'.
  - (ii) $\langle D\rangle^{BNC}\psi \in n'$ (satisfaction of $\langle D\rangle\psi$ has been postponed). We do not

add any vertex to the $D_{\sqsubset}$-structure, but simply proceed by taking into consideration the node n' and the current vertex v.

- *The Exit Beginning/Ending Cluster Rule has been applied.* Since $\mathcal{T}$ is open, the unique successor n' of n is open and it is the root of an irreflexive macronode. We add a new irreflexive vertex v' to V and an edge (v, v') to E. Moreover, we set the labeling of v' as the set of formulas belonging to n'. Then, we proceed by taking into consideration the node n' with the corresponding vertex v'.

To keep the construction finite, whenever the procedure reaches a tableau node n' that has been already taken into consideration, instead of adding a new vertex to the $D_{\sqsubset}$-structure, it simply adds an edge from the current vertex v to the vertex v' corresponding to n'.

Since any tableau for φ is finite, such a construction is terminating. However, the resulting structure $\langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ is not necessarily a $D_{\sqsubset}$-structure: there may exist a vertex $v \in V$ and a non-temporal formula $\psi \in \mathrm{CL}(\varphi)$ such that neither ψ nor $\neg\psi$ belongs to $\mathcal{L}(v)$. To overcome this problem, we can consistently extend the labeling $\mathcal{L}(v)$ as follows:

- if $\psi = p$, with $p \in \mathcal{AP}$, we put $\neg p \in \mathcal{L}(v)$;
- If $\psi = \neg\xi$, we put $\psi \in \mathcal{L}(v)$ if and only if $\xi \notin \mathcal{L}(v)$;
- If $\psi = \psi_1 \vee \psi_2$, we put $\psi_1 \vee \psi_2 \in \mathcal{L}(v)$ if and only if $\psi_1 \in \mathcal{L}(v)$ or $\psi_2 \in \mathcal{L}(v)$.

The resulting $D_{\sqsubset}$-structure $\langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ is a fulfilling $D_{\sqsubset}$-structure for φ and thus φ is satisfiable. □

Theorem 3.4 (COMPLETENESS) *Let φ be a $D_{\sqsubset}$-formula. If φ is satisfiable, then there exists an open tableau for it.*

Proof Let $\mathbf{S} = \langle\langle V, E\rangle, \mathcal{L}, \mathcal{B}, \mathcal{E}\rangle$ be a fulfilling $D_{\sqsubset}$-structure that satisfies φ. We take advantage of such a structure to show that there exists an open tableau $\mathcal{T}$ for φ. In particular, we will define a correspondence between (some) nodes in $\mathcal{T}$ and vertices in $\mathbf{S}$ that satisfies the following constraints:

(1) if n is associated with an irreflexive vertex v, then n belongs to an irreflexive macronode;

(2) if n is associated with a reflexive vertex v, then n belongs to a reflexive macronode;

(3) ff n is associated with a vertex v, then, for every formula $\psi \in n$, $\psi \in \mathcal{L}(v)$.

Let n_0 be the root of the tableau. We associate it with the root v_0 of $\mathbf{S}$. Since n_0 belongs to an irreflexive macronode, v_0 is an irreflexive vertex, and $\varphi \in \mathcal{L}(v_0)$, all constraints are satisfied.

Let n be the current node of the tableau, v be the vertex of $\mathbf{S}$ associated with it, and, by inductive hypothesis, n and v satisfy the constraints. We proceed by taking into consideration the rule that, according to the expansion strategy, is applicable to node n.

- *One of the Boolean Rules is applicable.* We consider the application of the

OR Rule to a formula of the form $\psi_1 \vee \psi_2$ (the other cases are simpler and thus omitted). Since $\psi_1 \vee \psi_2 \in n$, by Constraint (3), $\psi_1 \vee \psi_2 \in \mathcal{L}(v)$ and thus $\psi_1 \in \mathcal{L}(v)$ or $\psi_2 \in \mathcal{L}(v)$. If $\psi_1 \in \mathcal{L}(v)$, then we associate the successor n_1 of n, that contains ψ_1, with v; otherwise, we associate the successor n_2 of n, that contains ψ_2, with v. In either cases, all constraints are satisfied.

- *The Completion Rule is applicable.* Let us consider the application of the Completion Rule to the formula $\langle D\rangle\psi$. Since $\mathcal{L}(v)$ is an atom, either $\langle D\rangle\psi \in \mathcal{L}(v)$ or $[D]\neg\psi \in \mathcal{L}(v)$. In the former case, we associate the successor n_1 of n, that contains $\langle D\rangle\psi$, with v; in the latter case, we associate the successor n_2 of n, containing $[D]\neg\psi$, with v. In either cases, all constraints are satisfied.
- *The Marking Rule is applicable.* Let us consider the application of the Marking Rule to the formula $\langle D\rangle\psi$. According to the expansion strategy, n belongs to an irreflexive macronode and thus, by inductive hypothesis, v is an irreflexive vertex. Let $\mathcal{C}_b$ be the beginning successor cluster of v, $\mathcal{C}_e$ the ending successor cluster of v, and v_c their common reflexive successor (see Definition 2.3). Four cases may arise:
  (i) $\langle D\rangle\psi$ appears in $\mathcal{C}_b$, but not in $\mathcal{C}_e$ and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^B\psi$, with v.
  (ii) $\langle D\rangle\psi$ appears in $\mathcal{C}_e$, but not in $\mathcal{C}_b$ and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^E\psi$, with v.
  (iii) $\langle D\rangle\psi$ appears in $\mathcal{C}_b$ and $\mathcal{C}_e$, but not in v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^{BE}\psi$, with v.
  (iv) $\langle D\rangle\psi$ appears in $\mathcal{C}_b$, $\mathcal{C}_e$, and v_c. In this case, we associate the successor n' of n, which includes $\langle D\rangle^M\psi$, with v.
- *The Persistent Beginning/Ending Rule is applicable.* We associate the unique successor n' of n with v.
- *The Reflexive Step Rule is applicable.* According to the expansion strategy, n belongs to an irreflexive macronode and thus, by inductive hypothesis, v is an irreflexive vertex. Let v_b be a node in the beginning successor cluster of v, v_e a node in the ending successor cluster of v, and v_c the common reflexive successor of the two clusters. According to the expansion strategy, when such a rule turns out to be applicable, all $\langle D\rangle$-formulas have already been marked in accordance with $\mathbf{S}$. Let $n = \{\Phi, \langle D\rangle^B\Gamma, \langle D\rangle^M\mathrm{M}, \langle D\rangle^{BE}\Theta, \langle D\rangle^E\Lambda, [D]\Delta\}$, where Φ only contains atomic formulas. We have that $\{\langle D\rangle\Gamma, \langle D\rangle\Theta, \langle D\rangle\mathrm{M}, [D]\neg\Lambda, [D]\Delta, \neg\Lambda, \Delta\} \subseteq \mathcal{L}(v_b)$, that $\{\langle D\rangle\Lambda, \langle D\rangle\Theta, \langle D\rangle\mathrm{M}, [D]\neg\Gamma, [D]\Delta, \neg\Gamma, \Delta\} \subseteq \mathcal{L}(v_e)$, and that $\{\langle D\rangle\mathrm{M}, [D]\neg\Gamma, [D]\neg\Theta[D]\neg\Lambda, [D]\Delta, \neg\Gamma, \neg\Theta, \neg\Lambda, \Delta\} \subseteq \mathcal{L}(v_c)$. We associate the first successor of n with v_b, the second one with v_e, and the third one with v_c.
- *The Middle Step Rule is applicable.* According to the expansion strategy, n belongs to a macronode whose root is the middle node generated by an application of the Reflexive Step Rule and thus, by inductive hypothesis, n is associated with a middle reflexive vertex v_c. Since $\mathbf{S}$ is fulfilling, for every formula $\langle D\rangle\psi \in n$ there exists a successor v_ψ of v_c such that $\psi \in \mathcal{L}(v_\psi)$ and for

every $[D]\theta \in n$, $\theta, [D]\theta \in \mathcal{L}(v_\psi)$. For all $\langle D\rangle\psi \in n$, we associated the successor n_ψ of n with v_ψ.

- *The Build Beginning Cluster Rule is applicable.* Given the expansion strategy, by inductive hypothesis we have that n is associated with a node v that belongs to a beginning cluster $\mathcal{C}$. Let us consider the application of the rule to the formula $\langle D\rangle^B\psi$. Two cases may arise: either **S** fulfills $\langle D\rangle\psi$ outside $\mathcal{C}$ or not. In the former case, we associate the successor n' of n, that contains $\langle D\rangle^{BNC}\psi$, with v; in the latter case, there exists a node $v' \in \mathcal{C}$ such that $\psi \in \mathcal{L}(v')$ and we associate the successor n' of n, that contains both ψ and $\langle D\rangle^{BC}\psi$, with v'.
- *The Build Ending Cluster Rule is applicable.* This case is analogous to the previous one and thus omitted.
- *The Exit Beginning Cluster Rule is applicable.* Given the expansion strategy, by inductive hypothesis we have that n is associated with a node v that belongs to a beginning cluster $\mathcal{C}$. Let v' be the unique irreflexive successor of $\mathcal{C}$. We have that, for every formula $\langle D\rangle^{BNC}\psi \in n$, $\psi \in \mathcal{L}(v')$ or $\langle D\rangle\psi \in \mathcal{L}(v')$. The labeling of the unique successor node n' of n is thus consistent with v' and we can associate n' with v'.
- *The Exit Ending Cluster Rule is applicable.* This case is analogous to the previous one and thus omitted.

At the end of the above construction, we have obtained (a portion of) a tableau for φ. Since all its nodes are open, we can conclude that there exists an open tableau for φ. □

4 Conclusions

In [4], we devised a technique for constructing finite pseudo-models and building tableau-based decision procedures for logics of subinterval structures and applied it to the logic of strict subintervals. In this paper, we generalized it to the much more difficult case of the logic of proper subintervals. In such a way, we have completed the analysis and the proof of decidability for all versions of the semantics of subinterval logics (strict, proper, and reflexive) over dense linear orders, where point-intervals are not admitted. The inclusion of point-intervals is, however, unproblematic, because in the two difficult cases (strict and proper subinterval semantics) they are definable over dense linear orders by the formula $\langle D\rangle\bot$. Thus, the decidability results and tableau constructions carry over to subinterval structures with point-intervals after suitable minor modifications. On the contrary, the cases of discrete and arbitrary linear orders seem rather more difficult, and they are currently still under investigation.

Acknowledgement

This work has been funded by the bilateral project “Temporal logics in computer and information sciences”, supported by the Italian Ministero degli Affari Esteri and the National Research Foundation of South Africa, under the Joint Italy/South Africa Science and Technology Agreement. In addition, Davide Bresolin, Angelo Montanari, and Pietro Sala have been supported by the Italian PRIN project on "Constraints and preferences as a unifying formalism for system analysis and solution of real-life problems".

References

[1] A.A. Aaby and K.T. Narayana. Propositional temporal interval logic is PSPACE complete. In *Proc. of 9th Inter. Conf. Automated Deduction*, volume 193 of *LNCS*, pages 218–237. Springer, 1985.

[2] H. Bowman and S. Thompson. A tableau method for Interval Temporal Logic with projection. In *Proc. of TABLEAUX 1998*, volume 1397 of *LNAI*, pages 108–134. Springer, 1998.

[3] 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.

[4] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau Systems for Logics of Subinterval Structures over Dense Orderings. In N. Olivetti, editor, *Proc. of TABLEAUX 2007*, volume 4548 of *LNAI*, pages 73–89. Springer, 2007.

[5] 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.

[6] D. Bresolin and A. Montanari. A tableau-based decision procedure for a branching-time interval temporal logic. In H. Schlingloff, editor, *Proc. of the 4th Int. Workshop on Methods for Modalities (M4M)*, pages 38–53, 2005.

[7] D. Bresolin and A. Montanari. A tableau-based decision procedure for Right Propositional Neighborhood Logic. In B. Beckert, editor, *Proc. of TABLEAUX 2005*, volume 3702 of *LNAI*, pages 63–77. Springer, 2005.

[8] D. Bresolin, A. Montanari, and P. Sala. An optimal tableau-based decision algorithm for Propositional Neighborhood Logic. In *Proc. of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *LNCS*, pages 549–560. Springer, 2007.

[9] 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.

[10] Z. Chaochen, M.R. Hansen, and P. Sestoft. Decidability and undecidability results for duration calculus. In *Proc. of the 10th Symposium on Theoretical Aspects of Computer Science*, number 665 in LNCS, pages 58–68. Springer, 1993.

[11] L.K. Dillon, P.M. Melliar-Smith, L.E. Moser, G. Kutty, and Y.S. Ramakrishna. Interval logics and their decision procedures. Part I: an interval logic. *Theoretical Computer Science*, 166(1–2):1–47, 1996.

[12] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, October 1991.

[13] D. Van Hung and D.P. Guelev. Completeness and decidability of a fragment of duration calculus with iteration. In P.S. Thiagarajan and R. Yap, editors, *Advances in Computing Science*, volume 1742 of *LNCS*, pages 139–150. Springer, 1999.

[14] K. Lodaya. Sharpening the undecidability of interval temporal logic. In *Proc. of 6th Asian Computing Science Conference*, number 1961 in LNCS, pages 290–298. Springer, 2000.

[15] A. Montanari, G. Sciavicco, and N. Vitacolonna. Decidability of interval temporal logics over split-frames via granularity. In *Proc. the 8th European Conference on Logic in Artificial Intelligence (JELIA)*, number 2424 in LNAI, pages 259–270. Springer, 2002.