Artificial Intelligence 266 (2019) 1–26

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

On coarser interval temporal logics

Emilio Muñoz-Velasco [a], Mercedes Pelegrín [b], Pietro Sala [c], Guido Sciavicco [d,*], Ionel Eduard Stan [e]

[a] Department of Applied Mathematics, University of Malaga, Spain
[b] Department of Statistics and Operational Research, University of Murcia, Spain
[c] Department of Computer Science, University of Verona, Italy
[d] Department of Mathematics and Computer Science, University of Ferrara, Italy
[e] Department of Mathematics, Computer Science, and Physics, University of Udine, Italy

ARTICLE INFO

Article history:
Received 16 April 2017
Received in revised form 7 September 2018
Accepted 20 September 2018
Available online 17 October 2018

Keywords:
Modal and temporal logic
(Un)Decidability
Complexity

ABSTRACT

The primary characteristic of interval temporal logic is that intervals, rather than points, are taken as the primitive ontological entities. Given their generally bad computational behavior of interval temporal logics, several techniques exist to produce decidable and computationally affordable temporal logics based on intervals. In this paper we take inspiration from Golumbic and Shamir's coarser interval algebras, which generalize the classical Allen's Interval Algebra, in order to define two previously unknown variants of Halpern and Shoham's logic (HS) based on coarser relations. We prove that, perhaps surprisingly, the satisfiability problem for the coarsest of the two variants, namely HS_3, not only is decidable, but PSPACE-complete in the finite/discrete case, and PSPACE-hard in any other case; besides proving its complexity bounds, we implement a tableau-based satisfiability checker for it and test it against a systematically generated benchmark. Our results are strengthened by showing that not all coarser-than-Allen's relations are a guarantee of decidability, as we prove that the second variant, namely HS_7, remains undecidable in all interesting cases.

1. Introduction

Interval Temporal Logics (ITLs) consider time intervals as the primitive ontological entities. This represents an advantage when dealing with some relevant application domains such as planning and synthesis of controllers, which are characterized by advanced features that are neglected or dealt with in an unsatisfactory way by point-based formalisms. ITLs have been applied in several fields such as hardware and real-time system verification, language processing, constraint satisfaction and planning [38,2,21,40]. Moreover, due to the fact that temporal logics are considered as the natural basis for temporal extensions of Description Logics [6], several attempts have been made to design interval-based extensions of such formalisms [41,10,5,4]. ITLs can be also considered as the temporal counterpart of TSQL, that is, the temporal extension to the language SQL for databases, included in the standard SQL:2011 [45]. Halpern and Shoham's Modal Logic of Allen's Relations (HS) may very well be the most prominent ITL [28]. Its elegance and expressive power have attracted the attention of the temporal and modal logic communities; however, promising applications have been hampered by the fact, already discovered when

* Corresponding author.
E-mail addresses: ejmunoz@uma.es (E. Muñoz-Velasco), mariamercedes.pelegrin@um.es (M. Pelegrín), pietro.sala@univr.it (P. Sala), guido.sciavicco@unife.it (G. Sciavicco), stan.ioneleduard@spes.uniud.it (I.E. Stan).

https://doi.org/10.1016/j.artint.2018.09.001
0004-3702/

the logic was first introduced, that HS is highly undecidable. Various strategies have been considered in the literature to define fragments or variants of HS with a better computational behavior. These include constraining the underlying temporal structure [37], restricting the set of modal operators [14,1], softening the semantics to a reflexive one [30,33], restricting the nesting of modal operators [17], and restricting the propositional power of the languages [19].

Allen's Interval Algebra (IA) [2] is the backbone of HS: modal operators in the HS repertoire correspond to Allen's interval relations. Our proposal is based on Golumbic and Shamir's [26] idea to consider interval relations that describe a less precise relationship between intervals. They reduce the set of binary relations of Allen's Interval Algebra (IA) [2] by defining *coarser* relations, each corresponding to the logical disjunction of some Allen's relations; this approach generates two natural coarser algebras, namely IA_7 and IA_3. The former involves seven relations, by preserving the original relations *before*, *after*, and *equal to*, by joining *meets* and *overlaps* into a single relation (and similarly for their inverses), and by joining *during*, *starts*, and *finishes* into a single relation (and, again, similarly for their inverse ones). On the other hand, the latter takes into consideration only three relations: the original *before* and *after*, plus a relation (*intersects*) that can be viewed as the disjunction of all the remaining ones (and therefore is the inverse of itself and includes equality). The coarser algebras IA_7 and IA_3 inspire the logics HS_7 and HS_3 proposed in this paper. These languages follow similar ideas to the standard SQL:2011 [29], where interval relations are not necessarily Allen's ones (for example, *later* is interpreted as the disjunction of Allen's *meets* and *later*); therefore, they can be applied not only to classical areas of artificial intelligence, but also to temporal databases. We prove that the satisfiability problem for HS_3 is PSPACE-complete in the finite/discrete case, and it is PSPACE-hard when interpreted in any interesting class of linearly ordered sets, but that coarser relations do not guarantee the decidability of an interval temporal logic, because HS_7 remains undecidable over every interesting class of linearly ordered sets.

A partial picture of the computational behavior of the satisfiability problem for HS_3 in some cases could already be drawn from recent results concerning fragments of HS [34,15]. As a matter of fact, we can prove that HS_3 can be (polynomially) embedded into the $\mathsf{AB\overline{BA}}$ fragment of HS (the relations *before* and *after* can be immediately expressed in terms of *meets* and *met by*, while the relation *intersect* can be obtained by means of a combination of the modalities in $\mathsf{AB\overline{BA}}$); since $\mathsf{AB\overline{BA}}$ is decidable, but not primitive recursive, in the finite case [34] as well as in the case of the rational numbers [15], so is HS_3. Here we prove that its satisfiability problem is, in fact, PSPACE-complete (a much stronger result) in the finite/discrete case and PSPACE-hard in all other classes. Similarly, we know that just one modality in the HS_7 machinery, namely the one that corresponds to the disjunction of *starts, finishes* and *during*, alone implies the undecidability of the satisfiability problem of any temporal logic that contains it in the finite/discrete case [30]; here, we strengthen this results for HS_7, as our proof also applies to all cases already covered from [30]. It is worth mentioning that HS_3 is a unique case in the universe of the (few) decidable interval temporal logics; the only other example of interval temporal logics with a PSPACE satisfiability problem is the fragment $\mathsf{D\overline{D}B\overline{B}E\overline{E}}$ of HS, but only in the case of rational numbers [35], and the fragment D in its reflexive variant [33]. On the other hand, the maximal decidable fragments of HS in the finite/discrete case are the fragments $\mathsf{AB\overline{BL}}$ and $\mathsf{\overline{A}E\overline{E}L}$ (the fragment $\mathsf{AB\overline{BA}}$, that contains both, is decidable only on finite or rational frames), and they are not left/right symmetric, limiting their applicability (already hampered by the fact that their satisfiability problem is EXPSPACE-complete).

The paper is organized as follows. We first give the necessary preliminaries, and in Section 3 we give working examples that highlight the expressive power of HS_3. Section 4 is devoted to the complexity of the satisfiability problem for HS_3. In Section 5 we discuss the satisfiability problem for HS_7 and prove that it is generally undecidable, and, finally, in Section 6 we describe an implementation of a satisfiability checker for HS_3 along with the results of a series of systematic tests, before concluding in Section 7.

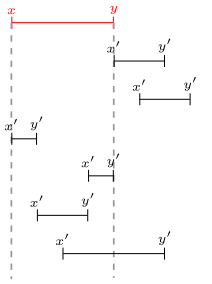
2. Preliminaries

In this section we give some necessary preliminaries for the rest of the work.

2.1. Syntax

Let $\mathbb{D} = \langle D, < \rangle$ be a strict linear order. A *strict interval* (respectively, *non-strict interval*) over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in D$ and $x < y$ (respectively, $x \leq y$). In the recent literature, the *strict semantics*, where only strict intervals are considered, is usually adopted. This conforms to the definition of interval adopted by Allen [2], but differs from the one given by Halpern and Shoham [28]. If we exclude the identity relation, there are 12 different relations between two intervals in a linear order, often called *Allen's relations* [2]: the six relations R_A (adjacent to), R_L (later than), R_B (begins), R_E (ends), R_D (during), and R_O (overlaps), depicted in Fig. 1, and their inverses, that is, $R_{\overline{X}} = (R_X)^{-1}$, for each $X \in \{A, L, B, E, D, O\}$. We interpret interval structures as Kripke structures, with Allen's relations playing the role of the accessibility relations. Thus, we associate a universal modality $[X]$ and an existential modality $\langle X \rangle$ with each Allen's relation R_X. For each $X \in \{A, L, B, E, D, O\}$, the *inverse* of the modalities $[X]$ and $\langle X \rangle$ are the modalities $[\overline{X}]$ and $\langle \overline{X} \rangle$, corresponding to the inverse relation $R_{\overline{X}}$ of R_X. Halpern and Shoham's logic HS [28] is a multi-modal logic with formulæ built from a finite, non-empty set $\mathcal{AP}$ of atomic propositions (also referred to as proposition letters), the classical propositional connectives, and a modal operator for each Allen's relation, as follows:

$$\varphi ::= \bot \mid p \mid \neg\psi \mid \psi \vee \xi \mid \psi \wedge \xi \mid \langle X \rangle \psi \mid \langle \overline{X} \rangle \psi.$$

HS	Allen's relations	Graphical representation
$\langle A \rangle$	$[x,y]R_A[x',y'] \Leftrightarrow y = x'$	
$\langle L \rangle$	$[x,y]R_L[x',y'] \Leftrightarrow y < x'$	
$\langle B \rangle$	$[x,y]R_B[x',y'] \Leftrightarrow x = x', y' < y$	
$\langle E \rangle$	$[x,y]R_E[x',y'] \Leftrightarrow y = y', x < x'$	
$\langle D \rangle$	$[x,y]R_D[x',y'] \Leftrightarrow x < x', y' < y$	
$\langle O \rangle$	$[x,y]R_O[x',y'] \Leftrightarrow x < x' < y < y'$	

$\mathrm{HS}_3/\mathrm{HS}_7$	Semantics
$\langle AO \rangle$	$\langle AO \rangle \equiv \langle A \rangle \vee \langle O \rangle$
$\langle DBE \rangle$	$\langle DBE \rangle \equiv \langle D \rangle \vee \langle B \rangle \vee \langle E \rangle$
$\langle I \rangle$	$\langle I \rangle \equiv \langle A \rangle \vee \langle \overline{A} \rangle \vee \langle O \rangle \vee \langle \overline{O} \rangle \vee \langle DBE \rangle \vee \langle \overline{DBE} \rangle$

Fig. 1. Allen's interval relations, HS modalities, and $\mathrm{HS}_3/\mathrm{HS}_7$ modalities.

In the above grammar, $p \in \mathcal{AP}$ and $X \in \{A, L, B, E, D, O\}$, and the other propositional connectives and constants (e.g., $\rightarrow$, and $\top$), as well as the dual modalities (e.g., $[A]\varphi \equiv \neg\langle A\rangle\neg\varphi$), can be derived in the standard way. In general, given any subset $S \subseteq \{X, \overline{X} : X \in \{A, L, B, E, D, O\}\}$, one can define the relation

$$R_S = \bigcup_{X \in S} R_X \cup \bigcup_{\overline{X} \in S} R_{\overline{X}}.$$

The corresponding modal operator can be denoted by simply juxtaposing the original symbols to obtain a string: for example, the modal operator that is the disjunction of Allen's relations *overlaps* and *during* would be denoted by $\langle O\,D\rangle$. In some cases, such as the relation *intersect*, we introduce a shorthand for the sake of readability, so that $I = A\overline{A}B\overline{B}E\overline{E}O\overline{O}D\overline{D}$. Well-formed HS_3 formulæ can be obtained by the above grammar with $X \in \{L, I\}$, while HS_7 formulæ are defined under the restriction that $X \in \{L, AO, DBE\}$.[1]

2.2. Semantics

The semantics of HS and both HS_3 and HS_7 is given in terms of *interval models* $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $\mathbb{D}$ is a linear order, $\mathbb{I}(\mathbb{D})$ is the set of all (strict) intervals over $\mathbb{D}$, and V is a *valuation function* $V : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})}$, which assigns to each atomic proposition $p \in \mathcal{AP}$ the set of intervals $V(p)$ on which p holds. The *truth* of a formula φ on a given interval $[x, y]$ in an interval model M is defined by structural induction on formulæ as follows:

- $M, [x, y] \Vdash p$ if $[x, y] \in V(p)$, for $p \in \mathcal{AP}$;
- $M, [x, y] \Vdash \neg\psi$ if $M, [x, y] \nVdash \psi$;
- $M, [x, y] \Vdash \psi \vee \xi$ if $M, [x, y] \Vdash \psi$ or $M, [x, y] \Vdash \xi$;
- $M, [x, y] \Vdash \psi \wedge \xi$ if $M, [x, y] \Vdash \psi$ and $M, [x, y] \Vdash \xi$;
- $M, [x, y] \Vdash \langle X \rangle \psi$ if there is $[z, t]$ with $[x, y] R_X [z, t]$ and $M, [z, t] \Vdash \psi$;
- $M, [x, y] \Vdash \langle \overline{X} \rangle \psi$ if there is $[z, t]$ with $[x, y] R_{\overline{X}} [z, t]$ and $M, [z, t] \Vdash \psi$.

Fig. 1 describes the semantics of HS_7 and HS_3 operators in terms of that of HS operators. Notice that a distinguishing characteristic ofs HS, inherited by the fragments considered in this paper, is the fact that the truth of a propositional letter over a given interval has no influence on the truth of the same propositional letter on the intervals contained in it, nor its points. Alternative choices include the *locality* principle, that implies assigning the same truth value to a propositional letter over an interval as over its starting points (see [38] for the introduction of locality in ITLs, and more recent work, such as [32], for an example of a renewed interest in constraining principles). This and other, similar, model-theoretic constraints have been shown useful in several applications and help reduce the complexity of problems such as satisfiability or model-checking, but here we follow the most general approach, in which undecidability is the rule and decidability the

[1] This notation should not be confused with the standard notation for fragments of HS, indicated by the set of its modal operators, e.g., $\mathsf{AB\overline{BA}}$, which includes four modal relations, namely, $\langle A \rangle$, $\langle \overline{A} \rangle$, $\langle B \rangle$, and $\langle \overline{B} \rangle$.

Table 1
Bike sharing system database.

Trips					
trip_id	starttime	stoptime	bikeid	from_st_id	to_st_id
4118	2013-06-27 12:11	2013-06-27 12:16	316	85	28
4275	2013-06-27 14:44	2013-06-27 14:45	64	32	32
4291	2013-06-27 14:58	2013-06-27 15:05	433	32	19
4316	2013-06-27 15:06	2013-06-27 15:09	123	19	19
4342	2013-06-27 15:13	2013-06-27 15:27	852	19	55
4480	2013-06-27 19:40	2013-06-27 22:28	27	340	46

Maintenance					
repair_id	starttime	stoptime	bikeid	from_st_id	to_st_id
5327	2013-06-28 09:05	2013-06-28 10:15	594	27	1
5335	2013-06-28 09:14	2013-06-28 10:41	227	26	1
5346	2013-06-28 09:26	2013-06-28 14:25	118	74	1
5353	2013-06-28 09:35	2013-06-28 09:50	226	24	1

exception. It is worth observing that both HS_3 and HS_7 are expressive enough to simulate the *universal* operator, and this depends essentially on their modalities being jointly exhaustive; for HS_3 we have:

$$[G]\varphi = \varphi \wedge \bigwedge_{X \in \{L,I\}} ([X]\varphi \wedge [\overline{X}]\varphi),$$

while for HS_7:

$$[G]\varphi = \varphi \wedge \bigwedge_{X \in \{L,AO,DBE\}} ([X]\varphi \wedge [\overline{X}]\varphi).$$

Formulæ of HS, and therefore of HS_3 and HS_7, can be interpreted over several different classes of interval models. Their frame properties sometimes influence the computational complexity of the satisfiability problem, as witnessed by the recent series of results [14,1]. Notable classes of linear orders include Lin, that is, the class of *all* linear orders, the class Fin of all *finite* linear orders, and the classes that are built over $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$. As far as the class of all dense linear orders is concerned, a simple application of both the downward Löwenheim–Skolem's Theorem and Cantor's Theorem proves that the decidability of a fragment of HS over $\mathbb{Q}$ implies the decidability of the same fragment on a generic dense linear order. We generically use the expression *the finite/discrete case* for the cases that include Fin, $\mathbb{N}$, and $\mathbb{Z}$, but also the class Dis of all discrete linear orders. Beside notable exceptions such as the fragment $\mathsf{ABB\overline{BA}}$ of HS, fragments of HS tend to behave in a similar way in all finite/discrete cases.

3. Coarser interval relations at work

In this section we discuss two working examples of temporal reasoning in coarser interval temporal logics.

3.1. An example as a specification language

This example is inspired by Divvy, the City of Chicago's bike sharing system. The system consists of a fleet of bikes that are locked into a network of docking stations located throughout the city. Bikes can be rented from and returned to any station in the city, creating a network of trips with many possible combinations of starting and ending points. Anonymous trip data are stored in a temporal database and openly available through the *Divvy Data Challenge* program. The database consists of two tables (see Table 1): *Trips*, that stores the data on the user rentals, and *Maintenance*, that stores the data on the repairs and other maintenance activities of the bikes. We assume that repairs and maintenance always take place at a special station with identifier 1. Since this is naturally temporal information, tables are equipped, among other attributes, with a *starttime* and a *stoptime* for each tuple; the interval identified by the two endpoints can be interpreted as the *valid time* of the tuple (see, e.g. [22]). Interval temporal logics can be used to reason on the database in at least two different ways: as query languages to extract information from a current instance of the data and as specification languages to express functional dependencies, integrity constraints and other requirements that must be met by every instance of the data; in this example, we follow the latter approach and use HS_3 as a specification language. Once the requirements are defined and formalized by the domain experts, one of the first problems that one must solve is the consistency problem, namely, the problem of checking whether they can be met by an actual instance of the database or not. In its most general formalization, the consistency problem can be solved by checking whether the set of formulæ representing the requirements is satisfiable.

Formulæ expressing the requirements will be built from the following propositional letters: $\mathit{trip}(i, j, k)$ to represent that the bike i performed a trip from station j to station k, and $\mathit{repair}(i, j, k)$ to represent that bike i was collected at station j and brought to station k to be repaired. Although HS_3 is not nearly as expressive as full HS, some very natural requirements

can be written in this language. To start with, we can enforce on the *Trip* table that at any time instant a given bike can belong to at most one trip. More formally, for every bike identifier i, no pairs of intervals satisfying $trip(i, j, k)$ and $trip(i, l, m)$ can share any point (in particular, a trip cannot start at the same time when the previous trip ended):

$$\bigwedge_{i,j,k}[G](trip(i, j, k) \rightarrow \bigwedge_{(j,k)\neq(l,m)} \neg trip(i, l, m)).$$

Similarly, we can ensure that two repairs for the same bike do not overlap:

$$\bigwedge_{i,j,k}[G](repair(i, j, k) \rightarrow \bigwedge_{(j,k)\neq(l,m)} \neg repair(i, l, m)).$$

Here and below, whenever necessary, we use $trip(i)$ as syntactic sugar for $\bigvee_{j,k} trip(i, j, k)$, and $repair(i)$ as syntactic sugar for $\bigvee_{j,k} repair(i, j, k)$:

$$\bigwedge_{i}[G](repair(i) \rightarrow (\neg\langle I\rangle repair(i))).$$

During maintenance a bike cannot do any trip (and vice-versa):

$$\bigwedge_{i}[G](repair(i) \rightarrow (\neg trip(i) \wedge \neg\langle I\rangle trip(i))),$$
$$\bigwedge_{i}[G](trip(i) \rightarrow (\neg repair(i) \wedge \neg\langle I\rangle repair(i))).$$

Then, we guarantee that a repair finishes in station 1 and starts from a station different from 1:

$$\bigwedge_{i,j}[G](\neg repair(i, 1, j) \wedge \bigwedge_{k\neq 1} \neg repair(i, j, k)).$$

Finally, we guarantee that maintenance is performed regularly:

$$\bigwedge_{i}[G]\langle L\rangle repair(i).$$

The above formalization is arguably simple and intuitive. While it may seem that a similar, maybe less immediate, formalization could be carried out in some point-based temporal logic such as LTL, it is easy to extend our model to include pure interval-based properties. Indeed, it is not uncommon that systems such as the above one are paired up with meteorological information, for statistical or knowledge extraction purposes. Moreover, such meteorological information can be enriched with data concerning the pollution levels, and, in particular, with anti-pollution measures taken by the city councils. Now, suppose that in order to promote the use of bike sharing systems over private cars it is decided that rates for those trips that occur during intervals of time in which anti-pollution measures are active must be discounted. In order to make the service more appealing, it is decided that such discounts apply to all trips that overlap in any way an interval of time in which anti-pollution measures are active. Assuming that *anti_pollution* labels all intervals in which some measure is active, and that $discount(i)$, as a shortcut for $\bigvee_{j,k} discount(i, j, k)$, labels all and only those trips that will be eligible for a discount, we can state:

$$\bigwedge_{i}[G]((trip(i) \wedge \langle I\rangle anti\_pollution) \rightarrow discount(i)).$$

3.2. An example in natural language processing

Natural language processing is a well-established branch of Artificial Intelligence, and it is well-known that the structure of natural language can be, at least partly, described with interval temporal logic constructs [40,3]. Recently, the rise of automatic personal assistants (e.g., *chatbots*) has given a new meaning to natural language processing, and tools for this task are being studied and developed. One of the major challenges in the automatic processing of a text is *context retrieval* [8], and, as a consequence, *context description*. A context can be seen as an interval during a conversation in which a particular topic is being discussed. Because of its nature, a context cannot be forced to be an uninterrupted sequence of instants in which that particular topic is being discussed; therefore, intervals are labeled with propositional letters that represent contexts without using any projection principles such as locality or homogeneity (see Section 2). Moreover, in a conversation the contexts may overlap each other, and more generally, they may be in any Allen's relation with each other.

In this example, a certain company is designing a natural language processing tool to analyze conversations between *agents* and *clients*. The agents contact the clients with the aim of selling a certain product, and the company has specific

requirements concerning the ordering and structure of the conversation, which can be described as requirements on the various contexts that arise during a conversation. They impose that some essential contexts, such as the price of the product (denoted by *price*), its known advantages (*advantages*) over other products, and its possible minor defects (*defects*) must arise during the conversation:

$$\langle L\rangle price \wedge \langle L\rangle advantages \wedge \langle L\rangle defects\,.$$

The company also wants that the disadvantages of the product are never discussed together with the price, in order to keep these two concepts separated during the conversation:

$$[G](price \rightarrow \neg\langle I\rangle defects\,).$$

In order to stress the positive message, the company also requires that the advantages are mentioned, separately, before and after mentioning the defects of the product:

$$[G](defects \rightarrow (\langle L\rangle advantages \wedge \langle \overline{L}\rangle advantages\,)).$$

Finally, the company wants that the price is never mentioned without recalling the advantages of the product:

$$[G](price \rightarrow \langle I\rangle advantages\,).$$

4. HS_3 is PSPACE-complete

In this section, we first prove that the satisfiability problem for HS_3 is PSPACE-hard, regardless of the class of linearly ordered sets on which interpreted, and, then, that it is also in PSPACE when it is interpreted in the class of all finite linear orders or in $\mathbb{N}/\mathbb{Z}$. As usual, the constructions and the underlying ideas in the finite case are emblematic for the entire range of classes of discrete structures; therefore, for the finite case we give a small model theorem in detail, a PSPACE, non-deterministic, algorithm, and a possible deterministic implementation in the form of a tableau, while for the case of $\mathbb{N}$ we simply outline a small periodic model theorem. The cases of $\mathbb{Z}$ and the class of all discrete linear orders Dis can be considered simple (albeit technically not trivial) generalizations of the case of $\mathbb{N}$.

4.1. Compass structure interpretation

Given a linear order $\mathbb{D}$ one can alternatively think of an HS model as a *compass structure* $\mathcal{G} = (\mathbb{D}, \mathcal{L})$, where strict intervals $[x, y]$ are seen as points (x, y) in the half-plane $\mathbb{D} \times \mathbb{D}$ identified by the constraint $x < y$, $\mathcal{L}$ is an *extended* labeling $\mathcal{L} : \mathbb{D} \times \mathbb{D} \rightarrow 2^{Cl(\varphi)}$, $Cl(\varphi)$ is the set of all sub-formulæ of a given formula φ, and $\mathcal{L}(x, y)$ denotes the subset of $Cl(\varphi)$ of precisely those formulæ that are true at the interval $[x, y]$ (including propositional letters). Modal operators are then immediately interpreted in a *geometric* way (e.g., the modality $\langle B\rangle$, $\langle \overline{B}\rangle$ correspond to moving on a vertical line in the plane, while $\langle E\rangle$, $\langle \overline{E}\rangle$ correspond to moving on a horizontal line). Such an interpretation, that works nicely also for its fragments of HS_3 and HS_7, was introduced in [44], and it has been used, among others, in [36,31], as an advanced tool for undecidability proofs.

4.2. Hardness

A Quantified Boolean (QB) formula is an expression of the form

$$\theta = Q_1 p_1 \ldots Q_n p_n f,$$

where f is a formula of propositional logic and, for all $1 \leq i \leq n$, Q_i is either $\forall$ or $\exists$. When every variable in a formula f is quantified, f is said to be a closed formula, and the *truth* problem for a closed QB formula is known to be PSPACE-hard [42]. In the following, we provide a (LOGSPACE) reduction from the truth problem for closed QB formulæ to the satisfiability problem for HS_3. Let $P_\forall$ (respectively, $P_\exists$) be the set of the indexes of the universally (respectively, existentially) quantified variables in θ. The strategy of our reduction is as follows. We want to encode a tree-like structure onto the linear ordering that underlies the model, without committing to any particular property of the ordering itself. Such a tree represents the QB formula θ in a very natural way: at each level $i > 1$, p_{i-1} occurs at least once if the i-th quantifier is existential (and it occurs either positive or negative), and at least twice if the i-th quantifier is universal (and it occurs both positively and negatively). So, for example, if f is a propositional formula where p_1 occurs, then the quantified formula $\forall p_1 f$ is encoded in a binary tree with empty root and two children: one with p_1 and the other one with $\neg p_1$. For a tree of height n (which corresponds to the encoding of a quantified formula with n propositions), we use $n+1$ propositional letters $h_1, \ldots, h_{n+1}$ to encode the levels of the (at most binary) tree. The idea of the construction is exemplified in Fig. 2, in which the colors are consistent with the intervals in which each letter h_i is true.

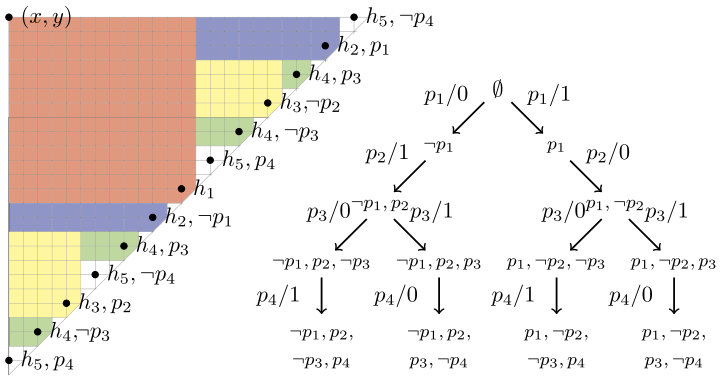


Fig. 2. A tree-model for $\theta = \forall p_1 \exists p_2 \forall p_3 \exists p_4((p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2) \wedge (p_3 \vee p_4) \wedge (\neg p_3 \vee \neg p_4))$ (right-hand side) and its embedding into a compass structure (left-hand side).

Let φ_{start} be the following formula:

$$\varphi_{start} = \quad h_1 \wedge \bigwedge_{1 \leq i \leq n+1} [G](h_i \rightarrow \neg \bigvee_{i < j \leq n+1} h_j)) \tag{1}$$

$$\wedge \bigwedge_{1 \leq i \leq n+1} [G](h_i \rightarrow \bigvee_{i < j \leq n+1} \neg \langle I \rangle h_j). \tag{2}$$

It is easy to see that, if $M, [x, y] \Vdash \varphi_{start}$, then: *(i)* $M, [x, y] \Vdash h_1$; *(ii)* if an interval $[z, t]$ satisfies h_i, then it does not satisfy any other h_j with $j \neq i$, and *(iii)* for every interval $[z, t]$, then there is no interval $[z', t']$ that intersects $[z, t]$ and satisfies any other h_j with $j \neq i$.

By interpreting the h intervals as nodes of the tree, we know, thanks to φ_{start}, that the root of the tree (h_1) exists. The idea is to encode f quantifier-by-quantifier, so that at each level we need to take care of the specific quantifier. The interval labeled by h_1 does not carry any propositional letter of f, and the truth values of $p_1, \ldots, p_n$ are encoded in h_i-intervals with $2 < i \leq n + 1$. In the following, we say that $[z', t']$ is a *descendant* of a given h_i-interval $[z, t]$ if one of two conditions apply: *(i)* either $t < z' < t' < s$ for each h_j-interval $[s, s']$ and each $j \leq i$ (*right* descendant), or *(ii)* $s < z' < t' < z$ for each h_j-interval $[s', s]$ and each $j \leq i$ (*left* descendant). Now, let φ_{quant} be the following formula:

$$\varphi_{quant} = \quad [G] \bigwedge_{i \in P_\forall} (h_i \rightarrow \langle I \rangle (\bigwedge_{1 \leq j < i} [I] \neg h_j \wedge \langle I \rangle (h_{i+1} \wedge p_i))) \tag{3}$$

$$\wedge [G] \bigwedge_{i \in P_\forall} (h_i \rightarrow \langle I \rangle (\bigwedge_{1 \leq j < i} [I] \neg h_j \wedge \langle I \rangle (h_{i+1} \wedge \neg p_i))) \tag{4}$$

$$\wedge [G] \bigwedge_{i \in P_\exists} (h_i \rightarrow \langle I \rangle (\bigwedge_{1 \leq j < i} [I] \neg h_j \wedge \langle I \rangle h_{i+1})). \tag{5}$$

Lemma 1. *Let* $M, [x, y] \Vdash \varphi_{start} \wedge \varphi_{quant}$.

(i) If $M, [z, t] \Vdash h_i$ *and the* i*-th quantifier of* f *is universal, then there exist distinct descendants* $[z', t']$ *and* $[z'', t'']$*, such that* $M, [z', t'] \Vdash h_{i+1} \wedge p_i$ *and* $M, [z'', t''] \Vdash h_{i+1} \wedge \neg p_i$*;*

(ii) If $M, [z, t] \Vdash h_i$ *and the* i*-th quantifier of* f *is existential, then there exists a descendant* $[z', t']$ *such that* $M, [z', t'] \Vdash h_{i+1} \wedge p_i$ *or* $M, [z', t'] \Vdash h_{i+1} \wedge \neg p_i$.

Proof. Suppose that $M, [x, y] \Vdash \varphi_{start} \wedge \varphi_{quant}$, that $M, [z, t] \Vdash h_i$, and that the i-th quantifier of f is universal. Then, (3) applies to $[z, t]$, and, therefore, there must be an $[z', t']$ such that $M, [z', t'] \Vdash h_{i+1} \wedge p_i$. At the same time, (4) applies, so that there must also be an $[z'', t']$ such that $M, [z'', t''] \Vdash h_{i+1} \wedge \neg p_i$; clearly, $[z', t'] \neq [z'', t'']$. Consider now the relative position of $[z', t']$ with respect to any other h_j-interval. First, notice that $[z', t']$ can neither coincide nor intersect $[z, t]$, thanks to φ_{start}. Therefore, either $z' > t$ or $t' < z$; let us assume the latter, without loss of generality. Let now be $[s', s]$ the h_j-interval, with $j = i - 1$ (if there is no such interval, then the position of $[z', t']$ is trivially correct) such that $s < z$, for each $s < s'' < z$, no interval of the type $[s''', s'']$ is an h_j-interval (in other words, $[s', s]$ can be regarded to as the immediate h_j-predecessor of $[z, t]$; such an intuitive notion, however, is imprecise over dense models). By φ_{start}, we know that $[z', t']$ can neither coincide nor intersect $[s', s]$. Towards a contradiction, suppose that $t < s'$, and that no interval $[u, u']$ with $s < u < u' < z$ is an h_{i+1}-interval. Then, (3) and (4) are both not satisfied; indeed, in this situation, every interval that intersects both $[z, t]$ and $[z', t']$ (and, so, every candidate to satisfy (3) or (4)) intersects also $[s', s]$, which is a contradiction.

Therefore, it must be the case that $s < z' < t' < z$, which is to say that $[z', t']$ is a descendant of $[z, t]$. Since the same argument can be applied to $[z'', t'']$, *(i)* follows. As for *(ii)*, one simply replicates the same argument as above, using (5), but only once. □

Finally, we need to make sure that once a decision over the truth value of a given p_i is taken, it is always respected. This means that the truth value of p_i on a given node in the tree must be copied over every node in its own subtree. Since the tree is encoded in a linear ordering, there must be a way to distinguish the *right* subtree of a node from the *left* one; this is not guaranteed from the encoding so far (i.e., point *(i)* of the above lemma may be satisfied by two intervals on the same side with respect to the current node); it is, however, a consequence of the remaining part of the construction, where φ_{casc} is the following formula:

$$\varphi_{casc} = [G] \bigwedge_{2 \le i \le n} ((h_i \wedge p_{i-1}) \rightarrow [I](\bigwedge_{1 \le j \le i-1} [I]\neg h_j \rightarrow [I]p_{i-1})) \tag{6}$$

$$\wedge [G] \bigwedge_{2 \le i \le n} ((h_i \wedge \neg p_{i-1}) \rightarrow [I](\bigwedge_{1 \le j \le i-1} [I]\neg h_j \rightarrow [I]\neg p_{i-1})). \tag{7}$$

Lemma 2. *Let $M, [x, y] \Vdash \varphi_{start} \wedge \varphi_{quant} \wedge \varphi_{casc}$, and let $M, [z, t] \Vdash h_i$ for some $2 \le i \le n + 1$.*

(i) If $M, [z, t] \Vdash p_{i-1}$, then each descendant of $[z, t]$ is labeled by p_{i-1};
(ii) If $M, [z, t] \Vdash \neg p_{i-1}$, then each descendant of $[z, t]$ is labeled by $\neg p_{i-1}$.

Proof. If $M, [x, y] \Vdash \varphi_{start} \wedge \varphi_{quant} \wedge \varphi_{casc}$, $M, [z, t] \Vdash h_i$, and $M, [z, t] \Vdash p_i$, then (6) applies. It is immediate to see that its consequent captures precisely the two maximal intervals[2] (one to the left and one to the right) such that each h-type interval that intersect either of them is a descendant of $[z, t]$ (or it is $[z, t]$ itself). Since these intervals are precisely where p_i is forced to hold, *(i)* is proved. The argument for *(ii)* is identical. □

Observe now that Lemma 1 and Lemma 2, combined, force any h_i-interval that correspond to a universal quantifier to have both a left and a right descendants, one with p_i and the other with $\neg p_i$, as we wanted. Let us now set

$$\varphi_\theta = [G](h_{n+1} \rightarrow f) \wedge \varphi_{start} \wedge \varphi_{quant} \wedge \varphi_{casc}.$$

Lemma 3. *Let $\theta = Q_1 p_1 \dots Q_n p_n f$ be a closed quantified Boolean formula. Then θ is true if and only if φ_θ is satisfiable.*

Proof. Suppose first that θ is true; we want to prove that φ_θ is satisfiable on a finite model. Let us proceed by induction on the number of variables of θ making sure that, at each step, the construction remains finite. As base case, suppose that θ has only one proposition, that is, $\theta = Q_1 p_1 f$, and f is a propositional formula in which only p_1 occurs. If Q_1 is universal, then f is satisfied no matter the value of p_1; we can easily build a model M over the domain $\mathbb{D} = \{0 < 1 < 2 < 3 < 4 < 5\}$, where $V(p_1) = V(h_2) = \{[0, 1], [4, 5]\}$ and $V(h_1) = \{[2, 3]\}$, and it is immediate to see that $M, [2, 3] \Vdash \varphi_{Q_1 p_1 f}$. The case in which Q_1 is existential can be treated in a similar way. As far as the inductive step is concerned, assume that $Q_1 p_1 Q_2 p_2 \dots Q_n p_n f$ is satisfiable. By the inductive hypothesis, we know for every quantified formula with $n - 1$ propositional variables, the corresponding HS_3 formula is satisfiable. Let us assume that Q_1 is universal. This means that the two formulæ $Q_2 p_2 \dots Q_n p_n f[p_1/0]$ and $Q_2 p_2 \dots Q_n p_n f[p_1/1]$ are satisfiable as well, and, since they have both $n - 1$ propositional variables, the inductive hypothesis applies. Let M_1 and M_2 be their respective models with finite domains $\mathbb{D}_1$ and $\mathbb{D}_2$. (Notice that we are assuming that M_1, M_2 are models over the letters $h_2, \dots, h_{n+1}, p_2, \dots, p_n$). Now, let M_1' (respectively, M_2') be the model obtained from M_1 (respectively, M_2) by simply making p_1 true (respectively, false) everywhere in the model. We now build a model M based on the domain $\mathbb{D}_1 \cup \{0 < 1\} \cup \mathbb{D}_2$, where we assume that $0, 1$ are new two fresh points and that 0 is greater than every point of $\mathbb{D}_1$ and, respectively, 1 is smaller than every point of $\mathbb{D}_2$. The valuation of M can be built as follows: *(i)* $V(h_1) = \{[0, 1]\}$; *(ii)* $V(p) = V_1'(p)$ for each interval that belongs to M_1' and each propositional variable; *(iii)* $V(p) = V_2'(p)$ for each interval that belongs to M_2' and each propositional variable; *(iv)* for each $[x, y] \neq [0, 1]$ such that $x \in \mathbb{D}_1 \cup \{0, 1\}$ and that $y \in \mathbb{D}_2 \cup \{0, 1\}$, $[x, y] \notin V(h_i)$ for any $1 \le i \le n$. It can be checked now that $M, [0, 1] \Vdash \varphi_{Q_1 p_1 Q_2 p_2 \dots Q_n p_n f}$ as we wanted. The case in which Q_1 is existential can be treated in a similar way.

Let us focus on the converse direction, and let us assume that φ_θ is satisfiable; we want to prove that θ is true. Let us proceed, as before, by induction on the number of propositional variables of θ. As base case, let $\theta = Q_1 p_1 f$, and let M be a linear model such that $M, [x, y] \Vdash \varphi_{Q_1 p_1 f}$; assume that Q_1 is universal. By, φ_{start}, Lemma 1, and Lemma 2, we know that $M, [x, y] \Vdash h_1$, that there exist two descendants $[x', y']$ and $[x'', y'']$ such that either $y' < x$ and $x'' > y$, or $x' > y$ and $y'' < x$, and that one of the following two cases hold: either $M, [x', y'] \Vdash h_2 \wedge p_1 \wedge f$ and $M, [x'', y''] \Vdash h_2 \wedge \neg p_1 \wedge f$, or $M, [x', y'] \Vdash$

[2] Over dense linear orders such intervals may not belong to the model, but the argument still holds.

$h_2 \wedge \neg p_1 \wedge f$ and $M, [x'', y''] \Vdash h_2 \wedge p_1 \wedge f$. Either way, f is satisfied no matter the value of p_1, so that $Q_1 p_1 f$ is true. The case in which Q_1 is existential can be treated in a similar way. As far as the inductive case is concerned, assume that $\theta = Q_1 p_1 Q_2 p_2 \dots Q_n p_n f$ and that Q_1 is universal. Since φ_θ is satisfiable by the hypothesis, for some linear model M and some interval $[x, y]$, we have that $M, [x, y] \Vdash \varphi_\theta$. Let $\mathbb{D}$ be the linear ordering on which M is based. By φ_{start}, $M, [x, y] \Vdash h_1$. Let $\mathbb{D}_1 \subset \mathbb{D}$ be set of points such that for each interval $[x', y']$ such that $x', y' \in \mathbb{D}_1$, $[x', y']$ is a left descendant of $[x, y]$, and let $\mathbb{D}_2 \subset \mathbb{D}$ be set of points such that for each interval $[x'', y'']$ such that $x'', y'' \in \mathbb{D}_1$, $[x'', y'']$ is a right descendant of $[x, y]$. By combining Lemmas 1 and 2 we know that $\mathbb{D}_1, \mathbb{D}_2$ are not empty, and that either $M, [x', y'] \Vdash p_1$ and $M, [x'', y''] \Vdash \neg p_1$ for each $[x', y'] \in \mathbb{I}(\mathbb{D}_1)$ and each $[x'', y''] \in \mathbb{I}(\mathbb{D}_2)$, or $M, [x', y'] \Vdash \neg p_1$ and $M, [x'', y''] \Vdash p_1$ for each $[x', y'] \in \mathbb{I}(\mathbb{D}_1)$ and each $[x'', y''] \in \mathbb{I}(\mathbb{D}_2)$. Notice that $\mathbb{D}_1$ (respectively, $\mathbb{D}_2$) can be seen as the underlying linear order of a model M_1 (respectively, M_2) for the propositional letters $h_2, \dots, h_{n+1}, p_2, \dots, p_n$. We also know that either $M_1, [x', y'] \Vdash \varphi_{Q_2 p_2 \dots Q_n p_n f[p_1/0]}$ for some $[x', y'] \in \mathbb{I}(\mathbb{D}_1)$ and that $M_1, [x'', y''] \Vdash \varphi_{Q_2 p_2 \dots Q_n p_n f[p_1/1]}$ for some $[x'', y''] \in \mathbb{I}(\mathbb{D}_2)$, or $M_1, [x', y'] \Vdash \varphi_{Q_2 p_2 \dots Q_n p_n f[p_1/1]}$ for some $[x', y'] \in \mathbb{I}(\mathbb{D}_1)$ and that $M_1, [x'', y''] \Vdash \varphi_{Q_2 p_2 \dots Q_n p_n f[p_1/0]}$ for some $[x'', y''] \in \mathbb{I}(\mathbb{D}_2)$. Thus, by inductive hypothesis, $Q_1 p_1 Q_2 p_2 \dots Q_n p_n f$ is true. The case in which Q_1 is existential can be treated in a similar way. □

Theorem 1. *The satisfiability problem for* HS_3 *interpreted over any class of linear orders is* PSPACE*-hard.*

Proof. Since Lemma 3 proves that the satisfiability problem for QB formulæ can be reduced in LOGSPACE to the satisfiability problem for HS_3, the result is immediate. □

4.3. PSPACE *membership: the finite case*

Now, we prove that the satisfiability problem for HS_3 is decidable in PSPACE in the particular case of finite linear orders.

We start by proving the *small model property* for HS_3 interpreted over finite models, which is key to showing decidability (and complexity) of the satisfiability problem. This is taken care by Lemma 4 below; given its importance, it is convenient to understand the driving concepts before getting into the technical details. In the finite case, every satisfiable formula has a finite model; we compute a bound for the size of such model, and we prove that every model exceeding the bound can be reduced to a smaller model that satisfies the same formula. To this end, we provide a description of each row of the compass structure that represents a model; such a (finite) description is carefully designed to achieve two goals: first, there must be a finite number of different descriptions, and, second, when two rows with the same description occur, the portion of the model above each of them must be (essentially) the same. So, the number of different descriptions will be our computed bound, and we shall refer to a model $\mathcal{G}$ (bigger than such a bound) and its contracted version $\mathcal{G}'$, where the portion of the model that exists between two rows with the same description is eliminated. The initial part of the proof is devoted to defining the description of a row; the second part to the definition of contracted model; and the last two parts of the proof to showing that in the resulting structure every universal formula is respected and every existential formula is eventually satisfied. Observe that each formula φ is equi-satisfiable with

$$\varphi' = \varphi \vee \langle I \rangle \varphi \vee \langle L \rangle \varphi.$$

Indeed, it is easy to see there exists a compass structure $\mathcal{G} = (\mathbb{D}, \mathcal{L})$ for φ if and only if there exists a compass structure $\mathcal{G}' = (\mathbb{D}, \mathcal{L}')$ for φ' with $\varphi' \in \mathcal{L}(0, 1)$; this notion is also called *initial satisfiability*. Then, without loss of generality, we focus on initial satisfiability only, that is, we search only for models whose domain is of the type $\{0, 1, 2, \dots\}$, where $[0, 1]$ is the initial interval.

Lemma 4. *Let* φ *be a finitely satisfiable* HS_3 *formula. Then it is satisfied on a model* M *with* $\mathbb{D}$ *of cardinality* $2^{O(|\varphi|^3)}$.

Proof. Let $\mathcal{G} = (\mathbb{D}, \mathcal{L})$ be a compass structure that (initially) satisfies the formula φ. We prove that if the size of $\mathbb{D}$ exceeds the bound, then there exists a compass structure $\mathcal{G}' = (\mathbb{D}', \mathcal{L}')$ for φ with $|\mathbb{D}'| < |\mathbb{D}|$. The *contraction* method obtained in this way can be iterated in order to obtain a model whose cardinality does not exceed the given bound.

Definitions. Given $\mathcal{G} = (\mathbb{D}, \mathcal{L})$ that satisfies φ, for a point (x, y) we can define the set of its *requests* as the subset of $Cl(\varphi)$ that holds on the interval $[x, y]$. Intuitively, a request is a temporal formula to be satisfied somewhere; it may be existential, and therefore must be witnessed by some interval, or universal, and therefore it influences only the labels of the already existing intervals. A set of requests behaves differently depending on the modal operators that are involved in it. We choose to consider the requests of the type $\langle I \rangle, \langle \overline{L} \rangle$ together, and separate them from the requests of the type $\langle L \rangle$, as follows: we define the set $\mathcal{R}_{I\overline{L}}(x, y)$ as the subset of $\mathcal{L}(x, y)$ that contains only formulas of the type $\langle I \rangle \psi, [I]\psi, \langle \overline{L} \rangle \psi$, or $[\overline{L}]\psi$, and the set $\mathcal{R}_L(x, y)$ as the subset of $\mathcal{L}(x, y)$ that contains only formulas of the type $\langle L \rangle \psi, [L]\psi$. By exploiting the properties of our modal operators we can represent the requests in a more convenient way. Indeed, observe that given two intervals $[x, y]$ and $[x', y]$ with $x > x'$, every existential request $\langle I \rangle \xi$ on $[x', y]$ is also on $[x, y]$, and every universal request $[I]\xi$ on $[x', y]$ is also on $[x, y]$; moreover, every existential request $\langle \overline{L} \rangle \xi$ on $[x, y]$ is also on $[x', y]$ and every universal request $[\overline{L}]\xi$ on $[x', y]$ is also on $[x, y]$. These two properties together allow us to define the set of requests of a *row* y in $\mathcal{G}$, as follows:

$$\mathcal{R}_{I\overline{L}}(y) = \{\mathcal{R}_{I\overline{L}}(x, y) : 0 \leq x < y\}.$$

Then, we can represent it as a sequence (which we call a *chain*) $R_1, R_2, \ldots, R_{|\mathcal{R}_{I\overline{L}}(y)|}$, because there is an implicit (set-containment) ordering among its elements. In a similar way, for every $0 \leq x, x' \leq y$, we have $\mathcal{R}_L(x, y) = \mathcal{R}_L(x', y)$ and, thus, on a row y, we can define:

$$\mathcal{R}_L(y) = \mathcal{R}_L(x, y), \text{for any } 0 \leq x < y.$$

(The asymmetry between types of requests is due to the inherent asymmetry of compass structures.) Now, observe that $|\mathcal{R}_{I\overline{L}}(x, y)|, |\mathcal{R}_L(x, y)| \leq |\varphi|$ for every interval $[x, y]$. This immediately implies that $|\mathcal{R}_L(y)| \leq |\varphi|$, but also that $|\mathcal{R}_{I\overline{L}}(y)| \leq |\varphi|$. To see this, recall that every element R of a chain $\mathcal{R}_{I\overline{L}}(y)$ can be ideally separated into four components, namely the existential and the universal requests of type $\langle I\rangle$ and $\langle\overline{L}\rangle$: the regularity property that we have observed above effectively limits the maximal cardinality of each component, and therefore also the cardinality of different Rs that may occur on a single row. As a consequence, the number of possible different chains $\mathcal{R}_{I\overline{L}}(y)$ is bounded by $|\varphi|! \leq |\varphi|^{|\varphi|} = 2^{|\varphi|\log(|\varphi|)}$. Now, if 0 is the first point of the model, then a chain $\mathcal{R}_{I\overline{L}}(y)$ contains sets of requests for the intervals $[0, y], [1, y], \ldots$, and so on; since, as we have already observed, the requests in (the components of the) successive Rs grow monotonically bigger or smaller, there will be a subset of points $0, \ldots, m \leq y$ in which all sets of requests are identical, and they change at the interval $[m + 1, y]$: this structure repeats itself along the chain, so that we can count, for each set of request that occurs in it, the number of consecutive points in which it appears. Now let $\mathcal{S} : \mathbb{I}(\mathbb{D}) \to 2^{Cl(\varphi)}$ be a function such that

$$\mathcal{S}(x, y) = \{\psi \in Cl(\varphi) : \text{ there are } x', y' \text{ such that } 0 \leq x' \leq x < y \leq y' \text{ and } \psi \in \mathcal{L}(x', y')\}.$$

Intuitively, $\mathcal{S}$ associates to each point (x, y) all the formulæ (in the closure of φ) that occur in the *maximum rectangle* that has (x, y) as it lower right corner (this notion is well defined as $\mathcal{G}$ is a finite compass structure and it is geometrically interpreted); we call $\mathcal{S}(x, y)$ the *rectangle of* (x, y). If we fix a row y, then it is easy to prove that the sets $\mathcal{S}(x, y)$ enjoy similar regularity properties as the sets of requests, so that $\mathcal{S}(0, y) \subseteq \mathcal{S}(1, y) \subseteq \ldots \subseteq \mathcal{S}(y - 1, y)$. As before, this allows us to effectively bound the cardinality of any such set; in fact, we have $|\mathcal{S}(x, y)| \leq 2 \cdot |\varphi|$. Therefore, we can represent these components on a row y as we did for the requests; if $\Psi(y) = \{\mathcal{S}(x, y) : 0 \leq x < y\}$ is the chain of sets $\mathcal{S}(x, y)$ on the row y, then $|\Psi(y)| \leq 2 \cdot |\varphi|$ (observe that the sets $\mathcal{S}(x, y)$ do not need to be consistent, so that it can contain both a formula and its negation). Each element of a chain $\Psi(y)$ can be chosen among $2^{2\cdot|\varphi|}$, so that the number of possible different chains $\Psi(y)$ is bounded by $2^{2\cdot|\varphi|} \cdot 2^{2\cdot|\varphi|-1} \cdot \ldots \cdot 1$, which, in turn, can be bounded by $2^{|\varphi|^2}$. Let us associate to a single row y a function

$$count(y) : \mathcal{R}_{I\overline{L}}(y) \times \Psi(y) \to \{1, \ldots, 4 \cdot |\varphi| + 1\},$$

such that, for each $R \in \mathcal{R}_{I\overline{L}}(y)$ and every $L \in \Psi(y)$, we have that

$$count(y)(R, L) = \min\{4 \cdot |\varphi| + 1, |\{x : \mathcal{R}_{I\overline{L}}(x, y) = R \text{ and } \mathcal{S}(x, y) = L\}|\}.$$

Observe that, for a row y, a chain $\mathcal{R}_{I\overline{L}}(y)$, and a chain $\Psi(y)$, there can be at most $(4 \cdot |\varphi| + 1)^{|\varphi|^2} = 2^{|\varphi|^2 \log(4\cdot|\varphi|+1)}$ possible $count(y)$ functions; this can be justified by observing that a given chain can be associated to a different function depending on how many times a given set of requests and how many times a given rectangle occur at a given position. At this point, for each y we define

$$row(y) = (\mathcal{R}_L(y), \mathcal{R}_{I\overline{L}}(y), count(y), \Psi(y)).$$

This can be regarded as a complete description of the row y of the compass structure (limited to the information that we need to perform the contraction). Taking into account the number of different component of each $row(y)$, the number of the possible values for $row(y)$ is bounded by

$$|\varphi| \cdot 2^{|\varphi|\cdot\log(|\varphi|)} \cdot 2^{|\varphi|^2\cdot\log(4\cdot|\varphi|+1)} \cdot 2^{|\varphi|^2} = |\varphi| \cdot 2^{|\varphi|\cdot(|\varphi|\cdot\log(4\cdot|\varphi|+1)+\log(|\varphi|)+|\varphi|)} = 2^{O(|\varphi|^3)},$$

as we wanted.

Contraction. If $|\mathbb{D}|$ exceeds the above bound, then, by a simple combinatorial argument, there must exist two rows $y_1 < y_2$ with $row(y_1) = row(y_2)$ (Fig. 3). Indeed, let us analyze the component of the above bound: $|\varphi|$ takes into account the number of different $\mathcal{R}_L(y)$ (in order for the contraction to work on the diagonal), $2^{|\varphi|\log(|\varphi|)}$ counts the number of possible different chains of requests (so that we make sure to have two rows with the same chain), $2^{|\varphi|^2\log(4\cdot|\varphi|+1)}$ counts the number of possible different functions $count(y)$ for the given row (so that we make sure that we have two row with the same chain and the same count), and $2^{|\varphi|^2}$ is the number of possible chains $\Psi(y)$ (to make sure that the rectangles identified by y_1 and y_2 is the same). In order to prove that we can safely contract the model between y_1 and y_2, let us focus our attention on the set $Rec(y_2)$ in $\mathcal{G}$ that contains all and only such points (z, t) where $z \leq y$ and $t > y$. For

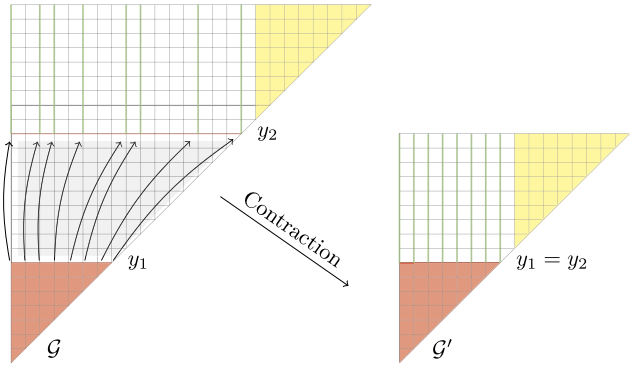


Fig. 3. A picture of the contraction method. In gray, the contracted area; in red, the down-left triangle (preserved); in yellow, the up-right triangle (moved without changes); and, in green, the vertical segments that have been copied from $\mathcal{G}$ to $\mathcal{G}'$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

each $\psi \in Cl(\varphi)$, if $\psi \in \mathcal{L}(z,t)$ and $(z,t) \in Rec(y)$, then we can identify four points (z_*, t_*) with $* \in \{right, left, up, down\}$ and $\psi \in \mathcal{L}(z_*, t_*)$ as follows: for every (z', t'), if $\psi \in \mathcal{L}(z', t')$ then $z_{left} \leq z' \leq z_{right}$ and $t_{down} \leq t' \leq t_{up}$. In other words, the points (z_*, t_*) with $* \in \{right, left, up, down\}$ are, respectively, the rightmost, leftmost, highest, and lowest occurrences of ψ in the rectangle of y. We define the set $\mathcal{W}(y) \subseteq \{0, \ldots, y\}$, for a given y, as a minimal subset of $\{0, \ldots, y\}$ such that contains all rightmost, leftmost, highest, and lowest witnesses of ψ in y for each $\psi \in Cl(\varphi)$. Thus, we can define the following non-decreasing monotone function:

$$g : \{0, \ldots, y_1\} \to \{0, \ldots, y_2\}.$$

In the above definition we have that *(i)* for every $x \in \{0, \ldots, y_1\}$, we have $\mathcal{R}_{I\overline{I}}(x, y_1) = \mathcal{R}_{I\overline{I}}(g(x), y_2)$, *(ii)* for every $x \in \{0, \ldots, y_1\}$, we have $\mathcal{S}(x, y_1) = \mathcal{S}(g(x), y_2)$, *(iii)* $\mathcal{W}_{y_2} \subseteq Img(g)$, and $y_2 - 1 \in Img(g)$, where $Img(g)$ is the image of g. Let $\Delta = y_2 - y_1$; we can finally build the compass structure $\mathcal{G}' = (\mathbb{D}', \mathcal{L}')$ with $|\mathbb{D}'| = |\mathbb{D}| - \Delta$, where $\mathcal{L}'$ is defined as follows:*(i)* $\mathcal{L}'(x, y) = \mathcal{L}(x, y)$ for every $0 \leq x \leq y \leq y_1$, *(ii)* $\mathcal{L}'(x, y) = \mathcal{L}(x + \Delta, y + \Delta)$ for every $y_1 < x \leq y \leq |\mathbb{D}'|$, and *(iii)* $\mathcal{L}'(x, y) = \mathcal{L}(g(x), y + \Delta)$ for every $y_1 < x \leq y_1 < y \leq |\mathbb{D}'|$. In the following, we shall denote the coordinates in $\mathcal{G}$ with $x, y, \ldots$, and those of $\mathcal{G}'$ with $x', y', \ldots$. There exists a natural correspondence between points in $\mathcal{G}$ and points in $\mathcal{G}'$, which depends on the particular area of $\mathcal{G}$ from which a certain point is copied; such a correspondence is key to the rest of the proof.

Consistency. Let us now prove that our new, smaller, compass structure is *consistent*, that is, that every universal request is respected. Given two points (x, y) and (z, t) in a compass structure, and given $X \in \{L, \overline{L}, I\}$, we say that $(x, y) \to_X (z, t)$ if and only if the following holds: if $(x, y) R_X (z, t)$ and $[X]\xi \in \mathcal{L}(x, y)$, then $\xi \in \mathcal{L}(z, t)$. In other words, $(x, y) \to_X (z, t)$ if and only if (z, t) is consistent with the universal requests of (x, y) along the relation X. The following cases may arise for two points $(x', y'), (z', t')$ in $\mathcal{G}'$.

- If $y', t' \leq y'_1$, then both points are located in an area of $\mathcal{G}'$ (the down-left triangle) that has been preserved after the contraction; therefore, $(x', y') \to_X (z', t')$ in $\mathcal{G}'$ for each $X \in \{L, \overline{L}, I\}$ thanks to the fact that $(x, y) \to_X (z, t)$ in $\mathcal{G}$.
- If $x', z' \geq y'_1$, then both points are located in an area of $\mathcal{G}'$ (the up-right triangle) that has been moved without changing its structure, that is, $x' = x + \Delta$ and $y' = y + \Delta$; therefore, $(x', y') \to_X (z', t')$ in $\mathcal{G}'$ for each $X \in \{L, \overline{L}, I\}$ thanks to the fact that $(x, y) \to_X (z, t)$ in $\mathcal{G}$.
- If $y' \leq y'_1$ and $z' \geq y'_1$, then clearly (x', y') belongs to the down-left triangle and two sub-cases arise: $y' < z'$, or $y' = y'_1 = z'$. In the first case, $(x', y') R_L (z', t')$ in $\mathcal{G}'$; this means that $(x, y) R_L (z + \Delta, t + \Delta)$, and, since $(x, y) \to_L (z + \Delta, t + \Delta)$ in $\mathcal{G}$, we immediately obtain that $(x', y') \to_L (z', t')$ in $\mathcal{G}'$. In the second case, $(x', y') R_A (z', t')$, that is, $(x', y') R_I (z', t')$. By construction, $\mathcal{L}'(x', y'_1) = \mathcal{L}(g(x), y_2)$ and $\mathcal{L}'(y'_1, t') = \mathcal{L}(g(y_1), t' + \Delta)$. Observe that $(g(x), y_2) R_I (\bar{x}, \bar{y})$ in $\mathcal{G}$ for every $(\bar{x}, \bar{y}) \neq (g(x), y_2)$ with $\bar{y} \geq y_2$ and $\bar{x} \leq y_2$; therefore, in $\mathcal{G}$, we have that $g(y_1) \leq y_2$ and $t' + \Delta > y_2$, and this implies that $(g(x), y_2) \to_I (g(y_1), t' + \Delta)$ in $\mathcal{G}$, which, in turn, implies that $(x', y'_1) \to_I (y'_1, t')$ in $\mathcal{G}'$.
- The case $z' \leq y'_1$ and $y' \geq y'_1$ is symmetric to the above one.
- If $y' \leq y'_1$ and $z' < y'_1 < t'$ then, as before, (x', y') belongs to the down-left triangle; moreover, there exists $\bar{x} \leq y'_1$ such that $\mathcal{L}'(z', t') = \mathcal{L}(g(\bar{x}), t' + \Delta)$. Again, two sub-cases arise, namely: $y' < z'$ or $y' \geq z'$. In the first case, $(x', y') R_L (z', t')$ in $\mathcal{G}'$ and, then, since g is monotonic non-decreasing, $(x, y) R_L (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$; this means that $(x, y) \to_L (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$, which, in turn, implies that $(x', y') \to_L (z', t')$ in $\mathcal{G}'$. If, on the other hand, $y' \geq z'$, then $(x', y') R_I (z', t')$ in $\mathcal{G}'$ and there are two further sub-cases. If $(x, y) R_I (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$, then, as before, $(x, y) \to_I (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$ implies $(x', y') \to_I (z', t')$ in $\mathcal{G}'$. If, on the other hand, $(x, y) R_L (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$, then, by construction, $(x, y) R_I (\bar{x}, y_1)$ and $\mathcal{L}(\bar{x}, y_1) = \mathcal{L}(g(\bar{x}), y_2)$. Then, since $\mathcal{S}(\bar{x}, y_1) = \mathcal{S}(g(\bar{x}), y_2)$, we have that $(x, y) \to_I (g(\bar{x}), \bar{y})$ in $\mathcal{G}$ for every $\bar{y} \geq y_2$; in particular $(x, y) \to_I (g(\bar{x}), t' + \Delta)$, and thus, $(x', y') \to_I (z't')$ in $\mathcal{G}'$.

- The case $t' \leq y'_1$ and $x' < y'_1 < y'$ is symmetric to the one above.
- If $x' > y'_1$ and $z' < y'_1 < t'$ then, by construction, there exists $\bar{x} \leq y_1$ such that $\mathcal{L}'(z', t') = \mathcal{L}(g(\bar{x}), t' + \Delta)$. Once again, two sub-cases may arise. If $t' < x'$, then $(x', y')R_{\overline{L}}(z', t')$ in $\mathcal{G}'$, which implies that $(x, y)R_{\overline{L}}(g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$, and, since $(x, y) \rightarrow_{\overline{L}} (g(\bar{x}), t' + \Delta)$ in $\mathcal{G}$, we have that $(x', y') \rightarrow_{\overline{L}} (z', t')$. If, on the other hand, $t' > x'$, then $t' + \Delta > x' + \Delta$; so, since $(x, y) \rightarrow_I (t' + \Delta)$ in $\mathcal{G}$, we have that $\mathcal{L}'(x', y') \rightarrow_I \mathcal{L}(z', t')$.
- The case $z' > y'_1$ and $x' < y'_1 < y'$ is symmetric.
- If $z' < y'_1 < t'$ and $x' < y'_1 < y'$, then in this situation we have $(x', y')R_I(z', t')$, since y'_1 belongs to both intervals. Then, by construction, there exist $\bar{x}, \bar{\bar{x}} \leq y_1$ such that $\mathcal{L}'(x', y') = \mathcal{L}(g(\bar{x}), y' + \Delta)$ and $\mathcal{L}'(z', t') = \mathcal{L}(g(\bar{\bar{x}}), t' + \Delta)$. Since $(g(\bar{x}), y' + \Delta)R_I(g(\bar{\bar{x}}), t' + \Delta)$ in $\mathcal{G}$ (where y_2 is the shared point in such case), we have that $(g(\bar{x}), y' + \Delta) \rightarrow_I (g(\bar{\bar{x}}), t' + \Delta)$ in $\mathcal{G}$, which implies that $(x', y') \rightarrow_I (z', t')$ in $\mathcal{G}'$.

Fulfillingness. Finally, it remains to prove that $\mathcal{G}'$ is *fulfilling*, that is, that every existential request is eventually satisfied. Given $(x', y') \in \mathbb{I}(\mathbb{D}')$, three cases may arise, depending on where the existential request has originated:

- If $y' \leq y'_1$, then as far as the requests of the type $\langle \overline{L} \rangle$ in $\mathcal{L}'(x', y')$ are concerned, since these originate and are satisfied in the down-left triangle of the structure which has not been modified by the contraction, fulfilling is guaranteed. Consider now a request of the type $\langle L \rangle \psi \in \mathcal{L}'(x', y') = \mathcal{L}(x, y)$, for which, in $\mathcal{G}$, there exists (z, t) with $z > y$ such that $\psi \in \mathcal{L}(z, t)$; three sub-cases arise, depending on the position of (z, t) in the structure $\mathcal{G}$. If $t \leq y_1$, then $\psi \in \mathcal{L}'(z', t') = \mathcal{L}(z, t)$ (again, everything is confined to the down-left triangle). If $t > y_2$, then, by construction, $\psi \in \mathcal{L}(z - \Delta, t - \Delta)$ (in this case, fulfilling is guaranteed by the fact that the up-right triangle has been moved without changing its structure). If, finally, $z \leq y_1 < t$ then, since g is monotone non-decreasing, there exists $\bar{x}$ such that $x \leq \bar{x} \leq y_1$ and $\psi \in \mathcal{L}(g(\bar{}), \bar{y})$ for some $\bar{y} > y_2$ and $g(\bar{x})$ is the rightmost vertical segment starting at y_2 and exhibiting ψ; thus $\psi \in \mathcal{L}'(\bar{x}', \bar{y} - \Delta)$, which is in relation R_L with (x', y'). It is important to notice that if $z > y_1$, then we are back to the second sub-case, as (in $\mathcal{G}$) $\langle L \rangle \psi$ belongs to both $\mathcal{R}_L(y_1)$ and $\mathcal{R}_L(y_2)$, which must be equal. Now, let $\langle I \rangle \psi \in \mathcal{L}'(x', y')$; then, in $\mathcal{G}$, there exists (z, t) such that $(x, y)R_I(z, yt)$ and $\psi \in \mathcal{L}(z, t)$. Two cases may arise, depending on the position of (z, t). If $t \leq y_1$, then $\psi \in \mathcal{L}'(z', t') = \mathcal{L}(z, t)$ (again, everything is confined to the down-left triangle). If, on the other hand, $t > y_1$, then $z \leq x < y_1 < t$, given that (z, t) must intersect (x, y). Let $\bar{x} \leq y$ be the leftmost index for which a vertical segment (starting at y_1) starts that exhibits ψ: by construction, $g(\bar{x})$ is the index which starts the leftmost vertical segment (starting at y_2) that exhibits ψ. This means that there exists $\psi \in \mathcal{L}(g(\bar{x}), \bar{y})$ for some $\bar{y} > y_2$, and, thus, $\psi \in \mathcal{L}'(\bar{x}, \bar{y} - \Delta)$.
- If $x' > y'_1$, then $x' = x - \Delta$ and $y' = y - \Delta$. As far as the requests of the type $\langle L \rangle$ in $\mathcal{L}'(x', y')$ are concerned, since these are originated and satisfied in the up-right triangle of the structure which has been move without modifications to its structure, fulfilling is guaranteed. Consider now a request of the type $\langle \overline{L} \rangle \psi \in \mathcal{L}'(x', y')$ for which, in $\mathcal{G}$, there exists (z, y) with $t < x$ such that $\psi \in \mathcal{L}(z, t)$; three sub-cases arise, depending on the position of (z, t) in the structure $\mathcal{G}$. If $z \geq y_2$, then $\psi \in \mathcal{L}'(z - \Delta, t - \Delta)$ (again, everything is confined to the up-right triangle). If $t \leq y_2$, then either $\langle \overline{L} \rangle \psi$ belongs to some label $\mathcal{L}(\bar{\bar{x}}, y_2)$ or ψ belongs to some label $\mathcal{L}(\bar{\bar{x}}, y_2)$; either way, $\psi \in \mathcal{L}'(\bar{z}', \bar{y}')$ for some $\bar{z}', \bar{y}' \leq y'_1$ (notice that, since ψ occurs somewhere in the down-left triangle, it fulfills the request originated in (x', y') independently from its precise position). If, finally $z < y_2 < t$ then there exists $\bar{x} \leq y_2$ and $\bar{y} < x$ such that $\psi \in \mathcal{L}(\bar{x}, \bar{y})$ and $\bar{x}$ is the vertical segment (starting at y_2) that exhibits ψ at the lowest coordinate; since g is monotone non-decreasing, and since we may assume $\bar{x} \in Img(g)$, there exists $\bar{\bar{x}} \leq y_1$ such that $g(\bar{\bar{x}}) = \bar{x}$, and, thus, $\psi \in \mathcal{L}'(\bar{\bar{x}}', \bar{y} - \Delta)$ (in $\mathcal{G}'$). Now, let $\langle I \rangle \psi \in \mathcal{L}'(x', y')$, for which, in $\mathcal{G}$, there exists (z, t) with $(x, y)R_I(z, t)$ such that $\psi \in \mathcal{L}(z, t)$; two sub-cases arise, depending on the position of (z, t) in the structure $\mathcal{G}$. If $z \geq y_2$, then $\psi \in \mathcal{L}'(z - \Delta, t - \Delta) = \mathcal{L}(z', t')$ (one more time, everything is confined to the up-right triangle). If, on the other hand, $z < y_2$, then $t > x$ (since the intervals $[x, y]$ and $[z, t]$ must intersect each other). We have that there exists $\bar{x} \leq y_2$ such that the vertical segment that starts at y_2 exhibits ψ at the highest coordinate; since g is monotone non-decreasing, and since we may assume $\bar{x} \in Img(g)$, there exists $\bar{\bar{x}} \leq y_1$ such that $g(\bar{\bar{x}}) = \bar{x}$. Now, we know that, for some $\bar{y} \geq t$, $\psi \in \mathcal{L}(\bar{x}, \bar{y})$ (in $\mathcal{G}$); by construction, $\psi \in \mathcal{L}'(\bar{\bar{x}}', \bar{y} - \Delta)$ (in $\mathcal{G}'$) and $(\bar{\bar{x}}', \bar{y} - \Delta)R_I(x - \Delta, y - \Delta)$, proving that the request is fulfilled.
- If $x' \leq y'_1$ and $y' > y'_1$, then there exists $\bar{x} \leq y_1$ such that $\mathcal{L}'(x', y') = \mathcal{L}(g(\bar{x}), y + \Delta)$. Clearly, $\bar{x} \leq x$, as g is monotonic non-decreasing. Let $\langle \overline{L} \rangle \psi$ in $\mathcal{L}'(x', y')) = \mathcal{L}(g(\bar{x}), y + \Delta)$. Then, since $\mathcal{G}$ is fulfilling and $\mathcal{R}_{\overline{L}}(g(\bar{x}), y_2) = \mathcal{R}_{\overline{L}}(\bar{x}, y_1)$, there exists (z, t) with $t < y_1$ such that $\psi \in \mathcal{L}(z, t)$. By construction, $\mathcal{L}(z, t) = \mathcal{L}'(z', t')$ (since (z, t) belongs to the down-left triangle), and, in $\mathcal{G}'$, $(x', y')R_{\overline{L}}(z', t')$; therefore, the request is satisfied. Let $\langle L \rangle \psi$ be in $\mathcal{L}'(x', y') = \mathcal{L}(g(\bar{x}), y + \Delta)$. Then, since $\mathcal{G}$ is fulfilling, there exists (z, t) with $z > y + \Delta$ such that $\psi \in \mathcal{L}(z, t)$. By construction, $\mathcal{L}(z, t) = \mathcal{L}'(z - \Delta, t - \Delta)$ and $(x', y')R_L(z - \Delta, t - \Delta)$, and, thus, the request is satisfied. Finally, let $\langle I \rangle \psi$ be in $\mathcal{L}'(x', y') = \mathcal{L}(g(\bar{x}), y + \Delta)$. Then, since $\mathcal{G}$ is fulfilling, there exists (z, t) such that, in $\mathcal{G}$, $(g(\bar{x}), y + \Delta)R_I(z, t)$ for which $\psi \in \mathcal{L}(z, t)$. Three cases may arise, depending on the position of (z, t). If $t \leq y_1$, then $\mathcal{L}(z, t) = \mathcal{L}'(z', t')$ (that is, (z, t) is in the down-left triangle, preserved by the contraction), and, since g is monotonic non-decreasing, we have that $\bar{x} \leq g(\bar{x})$; therefore, $(x', y')R_I(z', t')$, and the request is satisfied. If $z > y_2$ (that is, (z, t) is in the up-right triangle) then $\mathcal{L}(z', t') = \mathcal{L}'(z - \Delta, t - \Delta)$, and, since $(x', y')R_I(z - \Delta, t - \Delta)$, the request is satisfied. If, on the other hand, $z \leq y_2 < y$ and none of the previous case applies, then, by construction, there exist $\bar{x} \leq y_1$ such that $\psi \in \mathcal{L}(g(\bar{x}), \bar{y})$ for some $\bar{y} > y_2$ and, since $\mathcal{L}(g(\bar{x}), \bar{y}) = \mathcal{L}'(\bar{x}', \bar{y} - \Delta)$ and $(x', y')R_I(\bar{x}', \bar{y} - \Delta)$, the request is satisfied.

Now, $\mathcal{G}'$ is a compass structure for φ, and it is consistent and fulfilling. Clearly, $|\mathbb{D}'| < |\mathbb{D}|$. If $|\mathbb{D}'|$ is less than the claimed bound, we are done; otherwise, we repeat the entire process. □

A *small model for* HS_3 is any model for a HS_3 formula φ whose domain contains at most $2^{O(|\varphi|^3)}$ points. Small models are, evidently, exponential in length; this means that Lemma 4 proves that HS_3 is decidable in non-deterministic exponential time. A little more work is needed to prove that, in fact, it is decidable in polynomial space. To this end, we now describe a non-deterministic algorithm that uses only polynomial space, and, then, we obtain the result thanks to Savitch's Theorem (PSPACE = NPSPACE – see, e.g., [39]).

The idea underlying our polynomial-space algorithm is that we can check the satisfiability of a HS_3 formula by non-deterministically generating a model within the limits of Lemma 4 in such a way that, at any given moment, at most two (compass) rows are kept in memory, so that a single row must contain enough information to allow us to generate the next one. To this end, we need to introduce some more notation and definitions. An *atom* is any non-empty set $F \subseteq Cl(\varphi)$ such that, for every $\psi \in Cl(\varphi)$, $\psi \in F$ if and only if $\neg\psi \notin F$, and, for every $\psi \vee \xi \in Cl(\varphi)$, either $\psi \in F$ or $\xi \in F$; in other words, an atom is a maximally consistent subset of $Cl(\varphi)$ (labels on a compass structure are, in fact, atoms). We denote by $\mathcal{A}(\varphi)$ the set of all possible atoms for φ. For $X \in \{L, \overline{L}, I\}$, we denote by $\mathcal{R}_X^\varphi$ the set that contains all and only existential requests (that is, formulæ of the type $\langle X\rangle\xi$) in $Cl(\varphi)$, and, given the set $\mathcal{R}_I^\varphi$ and any subset Ψ of $Cl(\varphi)$, we call *observable*, denoted by $\mathcal{O}(\Psi)$, the set $\Psi \cap \mathcal{R}_I^\varphi$. Intuitively, we are restricting our attention to the existential part of the requests, and, in particular, those along the relation R_I. In order to efficiently describe a row, we define a *counting tuple* as a triple (R, R^*, n) such that $R \subseteq \mathcal{R}_I^\varphi$, $R^* \subseteq R$ and $n \in \mathbb{N}^+$; a counting tuple can be seen as the realization of an element of $\mathcal{R}_{I\overline{I}}(y)$ along with its count (the number of times that it occurs in a row). Intuitively, R contains requests *yet to be satisfied* or *satisfied at the row* y, and R^* contains requests *already satisfied* (somewhere below y in the compass structure). For us, a *row abstraction* is a word

$$CR = (R_0, R_0^*, n_0)\ldots(R_m, R_m^*, n_m)$$

of counting tuples such that: *(i)* $(R_i, R_i^*) \neq (R_{i+1}, R_{i+1}^*)$; *(ii)* $R_i \supseteq R_{i+1}$ for every $i < m$, and *(iii)* $R_i^* \supseteq R_{i+1}^*$ for every $i < m$. The *size* of a given row abstraction CR is the number $|CR| = \sum_{0\leq i\leq m} n_i$, and we access its elements as follows: we write $CR[i]$ (respectively, $CR^*[i]$), for $0 \leq i \leq |CR|$, to indicate the set R_j (respectively, R_j^*) such that $\sum_{0<l<j} n_l < i \leq \sum_{0<l\leq j} n_l$. In the algorithm shown in Figs. 4 and 5, CR and CR' are row abstractions.

Consider now the algorithm in Fig. 4. Clearly, to guarantee the completeness of the approach *LIM* must take the value is set to the theoretical bound for the dimension of any possible finite model for a given formula φ, as explained in Lemma 4. Each **let** step must be considered as a non-deterministic step, in which the requested object is guessed; then, the algorithm verifies its properties, and if any given property is not respected, the algorithm rejects (so, our algorithm may be seen as a PSPACE verifier). The requests of the type $\langle L\rangle$ that φ may have are handled in the following way: $f_L(\langle L\rangle\psi)$ represents an interval $[z, t]$ on the model being built such that $[z, t]$ satisfies ψ and that no other interval $[z', t']$ does, for $z' > z$; requests of the type $\langle\overline{L}\rangle$ are handled in a symmetric way. So, our algorithm correctly guesses all requests of the type $\langle L\rangle/\langle\overline{L}\rangle$, and considers them satisfied within the limits given by f_L and $f_{\overline{L}}$; in the function *Initialize*, we write $\mathbb{I}(LIM)$ to denote all intervals that can be built in the set $\{0, \ldots, LIM\}$. Then, a correct initial atom (F) is guessed, and its properties must be checked (within the function *CheckFirstAtom*). Obviously, the checked formula φ must belong to F, which, in turn, cannot contain any request of the type $\langle\overline{L}\rangle$, because F holds on $[0, 1]$ (first line of the conditions for F). Moreover, the requests of type $\langle L\rangle$ that are in F are precisely those satisfied anywhere to the right of $[0, 1]$, as collected in the function f_L (second line), and F contains, among others, those formulæ ever requested that are satisfied nowhere after (or before) $[0, 1]$ (third line). Finally, to be coherent with $f_{\overline{L}}$, F contains no formulæ whose leftmost satisfying interval is not $[0, 1]$ (fourth line). Right before the **while** loop, we describe the initial part of the compass structure that represent the model being guessed: it has only one level corresponding to $y = 1$, the row abstraction CR of such line contains precisely one counting tuple formed by non-starred requests only (nothing can be satisfied below 1). We take into account any formula ever mentioned on a line in the component Ψ_{CR}, of the requests of the type $\langle L\rangle$ in the component L_{CR}, and of the current step (at this point, 0).

The compass structure is guessed row-by-row, starting with line 1 (that corresponds to the set of all intervals ending at the point 1). Therefore, it is easy to check if, at a given point y, we have already finished: it is enough to verify that the current row abstraction has no request of type $\langle L\rangle$ and that every request in every set R of a counting tuple is matched into its corresponding set R^*, and therefore already satisfied. This explains the first part in the **while** loop. At any given *step*, the row $step + 2$ is being guessed, and its abstraction is called CR' (with set of formulæ $\Psi_{CR'}$). The remaining part of the code deals with checking that CR' is a correct guess. To this end, we first set the set of its requests of type $\langle L\rangle$ as precisely those insisting on the last point of the line (the point row), and we use a temporary variable $\Psi_{CR'}^*$ initially set at $\emptyset$. Then, we check CR'. First, we check that the number of components of CR' is correct (that is, it is precisely one more than CR); then, we check CR' tuple-by-tuple. Consider, then, the problem of checking the correctness of the i-th tuple, that corresponds to the interval $[i, row]$. The temporary variable CR_{pre}^* takes the value of the set of requests already satisfied below (and it is empty when $i = row - 1$). The guess CR' is rejected if: either the set of requests of type $\langle I\rangle$ already satisfied is not equal to the set of those that were already satisfied at the same point in CR plus those that are being satisfied in the line

```
procedure HS3-Sat(φ, LIM)
begin
    Initialize(f_L, f_L̄, LIM)
    let F ∈ A(φ) such that CheckFirstAtom(F, f_L, f_L̄)
    CR ← ⟨(R_I^F, ∅, 1)⟩
    Ψ_CR ← F
    L_CR ← R_L^F
    step ← 0
    while step < LIM do
        if L_CR = ∅ and ∀i(0 ≤ i ≤ |CR| → CR[i] = CR*[i]) then
            return Yes
        row ← step + 2
        let CR′ be a row abstraction
        let Ψ_CR′ ⊆ Cl(φ)
        L_CR′ ← {ψ : [0, row] R_L f_L(ψ)}
        Ψ*_CR′ ← ∅,
        if |CR′| ≠ |CR| + 1 then
            return No
        for i = 0 ... |CR′| do
            if i = |CR′| then
                CR*_pre ← ∅
            else
                CR*_pre ← CR[i]*
            if CR′[i]* ≠ CR*_pre ∪ O(Ψ_CR′ ∪ Ψ_CR) then
                return No
            if i < |CR′| and CR′[i] ⋡ CR[i] then
                return No
            let F ∈ A(φ) such that CheckAtom(CR′, f_L, f_L̄)
            Ψ*_CR′ ← Ψ*_CR′ ∪ F
        if Ψ*_CR′ ≠ Ψ_CR′ then
            return No
        CR ← CR′
        Ψ_CR ← Ψ_CR′
        step ← step + 1
    return No
```

Fig. 4. A PSPACE-algorithm for HS_3 formulæ satisfiability checking.

```
procedure Initialize(f_L, f_L̄, LIM)
begin
    let L ⊆ R_L^φ
    let f_L : L → I(LIM − 1)
    let L̄ ⊆ R_L̄^φ
    let f_L̄ : L̄ → I(LIM − 1)
```

```
procedure CheckFirstAtom(φ, f_L, f_L̄)
begin
    if φ ∈ F ∧ R_L̄^F = ∅
     and R_L(F) = {ψ : [0, 1] R_L f_L(ψ)}
     and F ⊇ {ψ : f_L(ψ) = [0, 1] ∨ f_L̄(ψ) = [0, 1]}
     and F ∩ {ψ : ¬([0, 1] = f_L̄(ψ))} = ∅ then
        return True
    return False
```

```
procedure CheckAtom(CR, f_L, f_L̄)
begin
    if F ⊆ Ψ_CR
     and Req_I(F) = CR[i]
     and R_L̄^F = {ψ : [i, row] R_L̄ f^L(ψ)}
     and R_L^F = {ψ : [i, row] R_L f^L(ψ)}
     and F ⊇ {ψ : f^L(ψ) = [i, row] ∨ f^L̄(ψ) = [i, row]}
     and F ∩ {ψ : ¬([i, row] R← f^L(ψ))} = ∅
     and F ∩ {ψ : ¬([i, row] R→ f^L(ψ))} = ∅ then
        return True
    return False
```

Fig. 5. Auxiliary procedures for the algorithm in Fig. 4.

(that is, anywhere in a interval of the type $[x, row]$ – they all intersect the interval $[i, row]$), or the chain of requests in CR', at the component i in CR' is not a correct chain. If CR', at the i-th component, has passed all above tests, then it only remains to be seen (guessed) which atom may occupy the label for the interval $[i, row]$; this is one key point to keep the complexity of the algorithm in PSPACE: atoms are only checked to exist, but never stored. The conditions for such an atom F to exist are: *(i)* its formulæ are in $\Psi_{CR'}$; *(ii)* its requests of type $\langle I \rangle$ are those in the i-th component of CR'; *(iii)* its requests of type $\langle L \rangle$ (respectively, $\langle \overline{L} \rangle$) are precisely those formulæ whose leftmost (respectively, rightmost) satisfying interval is yet to be seen (respectively, has already been seen); *(iv)* its formulæ include those satisfied precisely at $[i, row]$ and never again to the left or to the right; *(v)* finally, its formulæ cannot contradict the functions f (e.g., F cannot witness a formula whose leftmost satisfying interval starts somewhere before i): for a better readability, we denote, in Fig. 4, by $R_{\leftarrow}$ the union of the identity relation between intervals plus the relation R_X for $X \in \{E, \overline{E}, \overline{D}, \overline{O}, \overline{A}, \overline{L}\}$, and by $R_{\rightarrow}$ the union of the identity relation between intervals plus the relation R_X for $X \in \{B, \overline{B}, \overline{D}, O, A, L\}$. Now, $\Psi^*_{CR'}$ certainly includes F, and having guessed an atom for each i, the algorithm compares $\Psi^*_{CR'}$ (the set of all formulæ that should be on CR') with $\Psi_{CR'}$, and rejects if they are different.

Theorem 2. *The finite satisfiability problem for* HS_3 *is* PSPACE*-complete.*

Proof. Our procedure requires to store only two counters (row and $step$) and at most two row abstractions at the same time; therefore, it suffices to prove that they can be represented in polynomial space with respect to $|\varphi|$. Since none of the counters may exceed $2^{O(|\varphi|^3)}$, clearly $K = O(|\varphi|^3)$ bits are enough to store each one of them. Now, consider a row abstraction

$$CR = (R_1, R_1^*, n_1), (R_2, R_2^*, n_2), \ldots$$

and analyze its components. By definition, $R_1 \supseteq R_2 \supseteq \ldots$, and since each R_i is a subset of $\mathcal{R}_I^{\varphi}$, in the worst case each R_i differs from R_{i-1} because it has precisely one formula less. Therefore, if we focus on the R component only, there can be at most $2 \cdot |\varphi|$ different elements in CR. The component R^* follows the same pattern (reversed), so that, in total, there can be at most $4 \cdot |\varphi|$ different counting triples in a row abstraction. Each one of them requires $(2 \cdot \log(|\varphi| + 1) + K)$ bits to be represented, and the entire row abstraction requires $(2 \cdot \log(|\varphi| + 1) + K) \cdot K$ bits. To such quantity, one has to add the space to store three instances of Ψ_{CR} (one for CR, one for CR', and one temporary instance), each one of which requiring $\log(2 \cdot |\varphi| + 1)$ bits, and the space for a single atom F, requiring, again, $\log(2 \cdot |\varphi| + 1)$ bits. Summing up, the algorithm in Fig. 4 works in polynomial space. Since Theorem 1 gives us a matching lower bound, we have the claim. □

Our PSPACE algorithm is impractical. Each non-deterministic choice must guess very complex objects, and atoms must be generated in full at the beginning of the computation, which, alone, is a very time-demanding step. In Section 6 we describe a possible deterministic implementation of a finite satisfiability checker for HS_3, along with the result of a systematic series of tests.

4.4. PSPACE *membership: the case of natural numbers*

The satisfiability problem for HS_3 in the (general) discrete case, i.e., without assuming the finiteness of the models, can be decided using essentially the same principles as in the finite case; we give here the details for the case of $\mathbb{N}$.

The most relevant difference between the finite case and the case of $\mathbb{N}$ lies in the problem of model representation; in order to define a suitable (finite) representation for infinite models based on the natural numbers, we make use of the technical machinery defined in Lemma 4. First, given any compass structure $\mathcal{G} = (\mathbb{D}, \mathcal{L})$ and any index $0 < \overline{y}$, we denote $\mathcal{G}|_{\overline{y}} = (\{y \in \mathbb{D} : y < \overline{y}\}, \mathcal{L}|_{\overline{y}})$, where $\mathcal{L}|_{\overline{y}}$ is the labeling function defined as $\mathcal{L}|_{\overline{y}}(x, y) = \mathcal{L}(x, y)$ for every $0 \le x < y \le \overline{y}$, the $\overline{y}$ *prefix of* $\mathcal{G}$. It is essential to understand that stepping from $\mathcal{G}$ to $\mathcal{G}|_{\overline{y}}$ may have the effect of changing the content of $row(y)$ for some $y \le \overline{y}$: indeed, requests that were satisfied in $\mathcal{G}$ might not be satisfied anymore in $\mathcal{G}|_{\overline{y}}$. Therefore, we denote by $row|_{\overline{y}}(y)$ the value of the function row at y in the reduced compass structure $\mathcal{G}|_{\overline{y}}$. A finite compass structure $\mathcal{G}$ based on a domain $\mathbb{D} = \{0, \ldots, y_{max}\}$ is said to be *pseudo-fulfilling* if there exist two points $y < \overline{y} < y_{max}$ such that *(i)* $row(y) = row(\overline{y})$, *(ii)* for every $0 \le x \le y$ and for every $\psi \in \mathcal{R}_I(x, y)$, there exists $0 \le x' \le y$ and $0 \le y' \le \overline{y}$ such that $\psi \in \mathcal{L}(x', y')$, *(iii)* for every $\psi \in \mathcal{R}_L(x, y)$ there exists $y < x' < y' < \overline{y}$ such that $\psi \in \mathcal{L}(x', y')$, and *(iv)* $row|_{\overline{y}}(y) = row(y)$. The key point is that a pseudo-fulfilling structure is self-contained, so that every existential request originated on intervals that end before or at y, are satisfied on intervals that end before or at $\overline{y}$; notice, however, that the position of y_{max} plays an essential role in satisfying the last condition: even if conditions from (i) to (iii) are satisfied, a structure is pseudo-fulfilling only if, ideally, the portion of the model between y and $\overline{y}$ can be replicated infinitely often. Proving that HS_3 is decidable in PSPACE when interpreted over $\mathbb{N}$ requires three steps: first, proving that a model exists if and only if a pseudo-fulfilling structure (with φ in the label of the interval $[0, 1]$) does, second, that we can effectively bound the maximum cardinality of a pseudo-fulfilling structure, and, third, that checking the existence of a pseudo-model can be done in PSPACE. The first two steps are dealt with in Lemma 5 below, which is focused on the precise bound given that proving that pseudo-fulfilling structures can be converted into models is not essentially different from the proof of Lemma 4 (although a notion of periodicity is involved); as far as the third step is concerned, a simple modification of the algorithm in Fig. 4 is sufficient.

Lemma 5. *Let φ be a* HS_3 *formula satisfiable over* $\mathbb{N}$. *Then there exists a pseudo-fulfilling structure for φ with domain $\mathbb{D}$ of cardinality* $2^{O(|\varphi|^3)}$.

Proof. Since φ is satisfiable over $\mathbb{N}$ there exists a compass structure $\mathcal{G} = (\mathbb{N}, \mathcal{L})$ for it. We want to prove that there exists a bounded pseudo-fulfilling structure that describes it.

Definitions. Since the set of possible different row descriptions is finite, we take the smallest $y \in \mathbb{N}$ such that for every $y' \in \mathbb{N}$ there exists $y'' > y'$ with $row(y) = row(y'')$ (that is, the smallest y such that its own description repeats infinitely often). Then, we choose $\overline{y} > y$ in such a way that for every $0 \le x \le y$ and for every $\psi \in \mathcal{S}(x, y)$ there exists $x' \le x$ and $y < y' \le \overline{y}$ with $\psi \in \mathcal{L}(x', y')$, and for every $\psi \in \mathcal{R}_L(y)$ there exists $y < x' < y' \le \overline{y}$ with $\psi \in \mathcal{L}(x', y')$. In an identical way, we can also choose $y_{max} > \overline{y}$ as the smallest index such that $0 \le x \le \overline{y}$ and for every $\psi \in \mathcal{S}(x, \overline{y})$ there exists $x' \le x$ and $\overline{y} < y' \le y_{max}$ with $\psi \in \mathcal{L}(x', y')$, and for every $\psi \in \mathcal{R}_L(\overline{y})$ there exists $\overline{y} < x' < y' \le y_{max}$ with $\psi \in \mathcal{L}(x', y')$. Observe that $row(y)|_{y_{max}} = row(y)$ and $row(\overline{y})|_{y_{max}} = row(\overline{y})$; so, $\mathcal{G}_{y_{max}}$ is a pseudo-fulfilling structure (and $\varphi \in \mathcal{L}(0, 1)$), and we have to prove that we can contract it within the given bound.

Contraction. Given two points $\hat{y}, \hat{y}'$, we define

$$Occ(\hat{y}, \hat{y}') = \{\psi : \text{ there are } x, y' \text{ such that } 0 \le x \le \hat{y} < y' < \hat{y}' \text{ and } \psi \in \mathcal{L}(x, y')\}.$$

By construction, $Occ(y, \overline{y}) = Occ(\overline{y}, y_{max})$. As in Lemma 4, we can identify, for each $\psi \in Occ(y, \overline{y})$ four points (z_*, t_*) with $* \in \{right, left, up, down\}$ and $\psi \in \mathcal{L}(z_*, t_*)$, $z_* \le y$, and $t_* \le \overline{y}$, as follows: for every (z', t'), if $\psi \in \mathcal{L}(z', t')$, $z' < y$, and $y \le t' < \overline{y}$, then $z_{left} \le z' \le z_{right}$ and $t_{down} \le t' \le t_{up}$. In other words, the points (z_*, t_*) with $* \in \{right, left, up, down\}$ are, respectively, the rightmost, leftmost, highest, and lowest witnesses of each formula in $Occ(y, \overline{y})$. We collect all coordinates of the type z_* so defined in a set $Ver(\psi, y, \overline{y})$, and all coordinates of the type t_* in a set $Hor(\psi, y, \overline{y})$. In a similar way, we define the sets $Ver(\psi, \overline{y}, y_{max})$ and $Hor(\psi, \overline{y})$ for each $\psi \in Occ(\overline{y}, y_{max})$. Clearly, for each $\hat{y}'$, and each pair $\psi, \hat{y}$, it is the case that both $|Ver(\psi, \hat{y}, \hat{y}')|, |Hor(\psi, \hat{y}, \hat{y}')| \le 4$. We can now identify a set $Vb \subseteq \{x : 0 \le x \le y_{max}\}$ of *blocked verticals*, which is a minimal set that satisfies the following conditions: *(i)* $Ver(\psi, y, \overline{y}) \cup Ver(\psi, \overline{y}, y_{max}) \subseteq Vb$ for every $\psi \in Occ(y, \overline{y})$; and, for every $(R, L) \in dom(count(y))$, *(ii)* if $count(y)(R, L) < 4 \cdot |\varphi| + 1$ then $|\{x \in Vb : \mathcal{R}_{I\overline{L}}(x, y) = R \text{ and } \mathcal{S}(x, y) = L\}| = |\{x \in Vb : \mathcal{R}_{I\overline{L}}(x, \overline{y}) = R \text{ and } \mathcal{S}(x, \overline{y}) = L\}| = count(\overline{y})(R, L)$, and *(iii)* if $count(y)(R, L) \ge 4 \cdot |\varphi| + 1$ then $|\{x \in Vb : \mathcal{R}_{I\overline{L}}(x, y) = R \text{ and } \mathcal{S}(x, y) = L\}| = |\{x \in Vb : \mathcal{R}_{I\overline{L}}(x, \overline{y}) = R \text{ and } \mathcal{S}(x, \overline{y}) = L\}| = 4 \cdot |\varphi| + 1$. Observe that $|Vb| < 2 \cdot |\varphi|(4 \cdot |\varphi| + 1)$. Moreover, we can identify a set $Hb \subseteq \{y' : 0 \le y \le y_{max}\}$ of *blocked horizontals*, which as a minimal set that satisfies the following conditions: *(i)* $Hor(\psi, y, \overline{y}) \cup rows(\psi, \overline{y}, y_{max}) \subseteq Hb$ for every $\psi \in Occ(y, \overline{y})$, and *(ii)* $\{0, y, \overline{y}, y_{max}\} \subseteq Hb$. Observe that $|Hb| \le 2 \cdot |\varphi| + 4$. For every row $0 \le y \le y_{max}$ we can now define an additional counting function $count^V(y) : \mathcal{R}_{I\overline{L}}(y) \times \Psi_y \to \{1, \ldots, 2 \cdot |\varphi| \cdot (4 \cdot |\varphi| + 1)\}$ such that, for every $(R, L) \in \mathcal{R}_{I\overline{L}}(y) \times \Psi_y$,

$$count^V(y)(R, L) = |\{x : \mathcal{R}_{I\overline{L}}(x, y) = R \text{ and } \mathcal{S}(x, y) = L\}|.$$

Similarly to Lemma 4, we conclude that the number of possible $count^V(y)$ functions is $(2 \cdot |\varphi|(4 \cdot |\varphi| + 1))^{|\varphi|^2} = 2^{|\varphi|^2 \cdot \log(2 \cdot |\varphi| \cdot (4 \cdot |\varphi| + 1))}$. Now, consider two rows $0 \le \hat{y} < \hat{y}' \le y_{max}$ such that for every pair $y' < y'' \in Hb$ we have $y'' < \hat{y}$ or $\hat{y}' < y''$ or $y' < \hat{y} < \hat{y}' < y''$, that is, consider two rows $\hat{y}$ and $\hat{y}'$ strictly contained between two consecutive elements in Hb. If $row(\hat{y}) = row(\hat{y}')$ and $count^V(\hat{y}) = count^V(\hat{y}')$, then we can directly apply Lemma 4 and obtain a smaller structure $\mathcal{G}'$, which is, still, pseudo-fulling. If $\mathcal{G}_{y_{max}}$ is based on a domain $\mathbb{D}$, then $\mathcal{G}$ is based on a domain $\mathbb{D}'$ such that $|\mathbb{D}'| = |\mathbb{D}| - (\hat{y}' - \hat{y})$. But two rows $\hat{y} < \hat{y}'$ with the required properties are guaranteed to exist whenever the number of rows strictly contained between any two consecutive elements $y' < y''$ in Hb is greater than or equal to

$$N_\Delta = |\varphi| \cdot 2^{|\varphi| \cdot (|\varphi| \cdot \log(2 \cdot |\varphi| \cdot (4 \cdot |\varphi| + 1)) + \log(4 \cdot |\varphi| + 1) + \log(|\varphi|) + |\varphi|)}.$$

We can then iterate the contraction until $y'' - y' < N_\Delta$ for every two consecutive elements $y' < y''$. The pseudo-fulfilling structure obtained at the end of this process has cardinality less than or equal to

$$|Hb| \cdot N_\Delta = (2 \cdot |\varphi| + 4) \cdot |\varphi| \cdot 2^{|\varphi| \cdot (|\varphi| \cdot \log(2 \cdot |\varphi| \cdot (4 \cdot |\varphi| + 1)) + \log(4 \cdot |\varphi| + 1) + \log(|\varphi|) + |\varphi|)} = 2^{O(|\varphi|^3)},$$

as we wanted.

The pseudo-fulfilling structure obtained in this way can be now transformed into a compass structure based on the natural numbers, and therefore a model based on the natural numbers, by applying essentially the same technique already seen in Lemma 4. □

Dealing with the satisfiability of formulæ of HS_3 interpreted over the set of the integers does not require any mathematical tool different from those used above; the notion of pseudo-fulfilling structures can be generalized to be symmetric on the interval $[0, 1]$, and its limits suitably re-computed. Then, Lemma 5 works in the same way to prove its existence. A similar argument can be used to show that deciding the satisfiability of a formula of HS_3 interpreted in the class Dis of all discrete linear orders has the same computational properties.

Theorem 3. *The satisfiability problem for* HS_3 *in the case of natural numbers, in the case of the integers, and in the case of* Dis *is* PSPACE*-complete.*

5. HS_7 is undecidable

In this section we show that the satisfiability problem for HS_7, interpreted in any class of linearly ordered set that contains at least one infinite order, is undecidable; all finite/discrete cases were already covered by the results in [30]. Undecidability is proven via a reduction from the so-called Octant Tiling Problem (OTP) (as, for example, in [13]). This is the problem of establishing whether a given finite set of tile types $\mathcal{T} = \{\tau_1, \ldots, \tau_N\}$ can tile the second octant of the integer plane $\mathcal{O} = \{(n, m) \in \mathbb{Z}^2 : 0 \leq n \leq m\}$. For every tile type $\tau \in \mathcal{T}$, let $right(\tau)$, $left(\tau)$, $up(\tau)$, and $down(\tau)$ be the colors of the corresponding sides of τ. To solve the problem, one must find a function $f : \mathcal{O} \to \mathcal{T}$ such that $right(f(n, m)) = left(f(n + 1, m))$ and $up(f(n, m)) = down(f(n, m + 1))$. A simple application of König Lemma and of the Compactness Theorem for first-order logic allows one to prove that the OTP is undecidable; a similar argument is used in [11] to prove the undecidability of the Quadrant Tiling Problem.

In the following we reduce the OTP to the satisfiability problem for HS_7. Our construction is similar to other undecidability reductions for interval temporal and spatial logics [31,13], which, however, must be adapted due to the loss of expressive power of coarser operators. The reduction exploits *(i)* a correspondence between the points (x, y) of the octant and a suitable set of *unit* intervals (u-intervals from now on) labeled by the propositional letter u; *(ii)* propositional variables to represent tiles in $\mathcal{T}$; *(iii)* the modal operators of HS_7 to enforce the constraints of the problem. The starting point of our construction consists of building an infinite sequence of u-intervals. Let $\varphi_{u_{chain}}$ be the following formula:

$$\varphi_{u_{chain}} = \quad u_0 \wedge \bigwedge_{l=0,1} [G](u_l \to \langle AO \rangle u_{(l+1) \bmod 2}) \tag{8}$$

$$\wedge [G](u \leftrightarrow (u_0 \vee u_1)) \tag{9}$$

$$\wedge [G](u \to [DBE]\neg u) \tag{10}$$

$$\wedge \bigwedge_{l=0,1} [G](u_l \to [DBE]u'_l) \tag{11}$$

$$\wedge [G]((u'_0 \wedge u'_1) \to \bot). \tag{12}$$

Lemma 6. *Let* $M, [x, y] \Vdash \varphi_{u_{chain}}$*. Then, there exists an infinite increasing sequence of points* $y_0, y_1, y_2, \ldots$ *such that*

(i) $y_0 = y$*;*
(ii) for each $i \geq 0$*,* $M, [y_i, y_{i+1}] \Vdash u$*;*
(iii) if $[z, t] \neq [y_i, y_{i+1}]$*, for each* $i \geq 0$*, then* $M, [z, t] \Vdash \neg u$*, unless* $t \leq x$ *or* $z > y_i$ *for each* $i \in \mathbb{N}$*.*

Proof. By hypothesis, $M, [x, y] \Vdash \varphi_{u_{chain}}$; therefore, $M, [x, y] \Vdash u_0$. Because of (8) there must be an interval $[y_0, y_1]$ such that $M, [y_0, y_1] \Vdash u_1$, and that $x < y_0 \leq y$. To prove *(i)* we suppose, for the sake of contradiction, that $y_0 < y$. By (11) we obtain $M, [y_0, y] \Vdash u'_0$ and also $M, [y_0, y] \Vdash u'_1$, contrary to (12). Consequently, $y = y_0$. Following the same argument and taking (9) into account, one immediately sees that $\varphi_{u_{chain}}$ forces the existence of a chain $y = y_0 < y_1 < y_2 \ldots$ of u-intervals verifying *(ii)*. Suppose now, for the sake of contradiction, that $M, [z, t] \Vdash u$ for some $[z, t] \neq [y_i, y_{i+1}]$ for all $i \in \mathbb{N}$, where $t > x$ and $z < y_i$ for some $i \in \mathbb{N}$. Let us analyze all possible cases that may occur. If $y_i \leq z < t \leq y_{i+1}$ for some $i \in \mathbb{N}$, or $x \leq z < t \leq y_0$, then we have a contradiction with (10), because $[z, t]$ is a u-interval contained (via R_{DBE}) in another u-interval, $[y_i, y_{i+1}]$ or $[x, y_0]$. If $y_i \leq z < y_{i+1}$ and $t \geq y_{i+k}$ for some $i, k \in \mathbb{N}$ and $k > 1$, or $x \leq z < y_0$ and $t \geq y_1$, then we have, again, a violation of (10). If $y_i \leq z < y_{i+1}$ and $y_{i+1} < t < y_{i+2}$, or $x \leq z < y_0$ and $y_0 < y < y_1$, then, thanks to (9), either $M, [z, t] \Vdash u_0$ or $M, [z, t] \Vdash u_1$. Assume that $[y_i, y_{i+1}]$ is a u_0-interval: in the first case, we have a contradiction with (12) because $[y_{i+1}, t]$ is both a u'_0-interval and a u'_1-interval, while in the second case we have a contradiction because $[z, y_{i+1}]$, in turn, is both a u'_0-interval and a u'_1-interval. If $[y_i, y_{i+1}]$ is a u_1-interval, then the symmetric argument applies. If $t > y_i$ for each $i \in \mathbb{N}$, then, since by the hypothesis $z < y_i$ for some $i \in \mathbb{N}$, we have a contradiction with (10). Finally, if $z < x$, then $t < y_0$ because of (10). Thanks to (9), $M, [z, t] \Vdash u_0$ or $M, [z, t] \Vdash u_1$. In the first case, (8) ensures the existence of a u_1-interval that will cause a conflict with the u_0-interval $[x, y_0]$. Otherwise, since $x < t$, the interval $[x, t]$ satisfies u'_0 and u'_1, which, again, is a contradiction. □

By the following formula, which we call $\varphi_{u_{be}}$, we can easily identify those intervals that *start* and those that *end* u-intervals:

$$\varphi_{u_{be}} = \quad [G](u \to (\langle DBE \rangle \top \wedge [DBE]u') \tag{13}$$

$$\wedge [G]((u' \wedge [AO]\neg u) \to u_b) \tag{14}$$

$$\wedge [G]((u' \wedge [\overline{AO}]\neg u) \rightarrow u_e) \tag{15}$$

$$\wedge [G](u_b \rightarrow [AO]\neg u) \tag{16}$$

$$\wedge [G](u_e \rightarrow [\overline{AO}]\neg u). \tag{17}$$

Lemma 7. *Let $M, [x, y] \Vdash \varphi_{u_{chain}} \wedge \varphi_{u_{be}}$, and let $y_0, y_1, y_2, \ldots$ be the infinite sequence of points whose existence is guaranteed by Lemma 6.*

(i) For each $i \geq 0$, every interval of the form $[y_i, z]$, where $z < y_{i+1}$, satisfies u_b but not u_e (and there exists at least one interval of this form);
(ii) For each $i \geq 0$, every interval of the form $[z, y_{i+1}]$, where $z > y_i$, satisfies u_e but not u_b (and there exists at least one interval of this form).

Proof. Consider, first, a u-interval $[y_i, y_{i+1}]$ of the sequence, and a point z such that $y_i < z < y_{i+1}$. Such z exists thanks to (13). Also by (13), the interval $[y_i, z]$ is certainly a u'-interval; since, by Lemma 6, that neither $[z, t]$ for any $z < t < y_{i+1}$, nor any $[t', t]$ for any $y_i < t' < z$, can be a u-interval, (14) applies, and $[y_i, z]$ must be a u_b-interval. Since $[y_{i-1}, y_i]$ is u-interval, (17) applies, and $[y_i, z]$ cannot be a u_e-interval, proving *(i)*. Point *(ii)* is proven likewise.

Consider now the following formula, in which $\mathfrak{L} = \{*\} \cup \mathcal{T}$ (we abuse of notation and identify tile symbols with the propositional letters that represent them):

$$\varphi_{conf} = \quad C_0 \wedge \bigwedge_{l=0,1} [G](C_l \rightarrow \langle AO\rangle C_{(l+1) \bmod 2}) \tag{18}$$

$$\wedge [G](C \leftrightarrow (C_0 \vee C_1)) \tag{19}$$

$$\wedge \bigwedge_{l=0,1} [G](C_l \rightarrow [DBE]C'_l) \tag{20}$$

$$\wedge [G]((C'_0 \wedge C'_1) \rightarrow \bot) \tag{21}$$

$$\wedge [G](C \rightarrow ([AO]\neg u_e \wedge [\overline{AO}]\neg u_b)) \tag{22}$$

$$\wedge [G](C \rightarrow ([DBE]\neg C \wedge \langle DBE\rangle \top)) \tag{23}$$

$$\wedge [G](u \leftrightarrow \bigvee_{s\in\mathfrak{L}} s)) \tag{24}$$

$$\wedge [G] \bigwedge_{s,s'\in\mathfrak{L}, s\neq s'} (s \wedge s' \rightarrow \bot) \tag{25}$$

$$\wedge \langle AO\rangle(* \wedge \langle AO\rangle(\bigvee_{\tau\in\mathcal{T}} \tau \wedge \langle AO\rangle *)) \tag{26}$$

$$\wedge [G]((u \wedge \langle AO\rangle C) \rightarrow \langle AO\rangle *) \tag{27}$$

$$\wedge [G]((u \wedge \langle \overline{AO}\rangle C) \rightarrow \langle \overline{AO}\rangle *) \tag{28}$$

$$\wedge [G](* \rightarrow (\langle AO\rangle C \vee \langle \overline{AO}\rangle C). \tag{29}$$

Lemma 8. *Let $M, [x, y] \Vdash \varphi_{u_{chain}} \wedge \varphi_{u_{be}} \wedge \varphi_{conf}$, and let $y_0, y_1, y_2, \ldots$ be the infinite sequence of points whose existence is guaranteed by Lemma 6. Then there exists an infinite sequence of indexes $k_0, k_1, k_2, \ldots$ such that*

(i) each u-interval of the type $[y_i, y_{i+1}]$ satisfies precisely one letter in $\mathfrak{L}$;
(ii) for each $j \geq 0$, $M, [y_{k_j}, y_{k_{j+1}}] \Vdash C$;
(iii) the C-interval $[y_{k_0}, y_{k_1}]$ is composed of exactly three units, the middle one of which is $\tau \in \mathcal{T}$;
(iv) if $[z, t] \neq [y_{k_j}, y_{k_{j+1}}]$ for each $j \geq 0$, then $M, [z, t] \Vdash \neg C$, unless $t \leq x$ or $z > y_i$ for each $i \in \mathbb{N}$;
(v) each C-interval of the type $[y_{k_j}, y_{k_{j+1}}]$ is such that both its first unit $[y_{k_j}, y_{k_j+1}]$ and its last unit $[y_{k_{j+1}-1}, y_{k_{j+1}}]$ satisfy $*$;
(vi) if $[z, t] \neq [y_{k_j}, y_{k_j+1}]$ and $[z, t] \neq [y_{k_{j+1}-1}, y_{k_{j+1}}]$ for each $j \geq 0$, then $M, [z, t] \Vdash \neg *$, unless $t \leq x$ or $z > y_i$ for each $i \in \mathbb{N}$.

Proof. Point *(i)* is an immediate consequence of (24) and (25). Since $M, [x, y] \Vdash C_0$ by (18), there must be an interval $[z, t]$ such that $x < z \leq y$ and $t > y$ such that $M, [z, t] \Vdash C_1$. Suppose, for the sake of contradiction, that $z < y$. By (20), we obtain $M, [z, y] \Vdash C'_0$ and also $M, [z, y] \Vdash C'_1$, contrary to (21). Therefore $z = y = y_0$, and we set $k_0 = 0$. Now we prove *(iii)*, i.e., $t = y_3 = y_{k_1}$. First, note that (22), in combination with Lemma 7, avoids that $y_i < t < y_{i+1}$ for any $i \in \mathbb{N}$. Formulæ (24) and (26) imply that $[y_0, y_1]$, $[y_1, y_2]$, and $[y_2, y_3]$ are, respectively, $*$-, τ-, and $*$-intervals, for some tile τ. Therefore, $t = y_1$

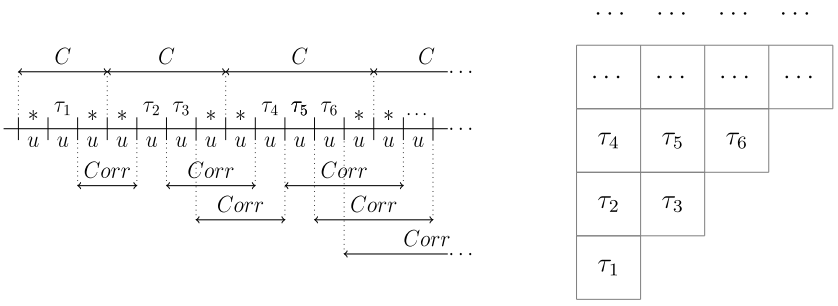


Fig. 6. A model of $\varphi_{\mathcal{T}}$ and its interpretation as a tiling of the second octant.

contradicts (27) and $t = y_2$ contradicts (28), because of *(i)*; applying (29), (19), (20), and (21) to $[y_2, y_3]$, it is easy to see that $t > y_3$ is not possible. Then, $y_{k_1} = y_3 = t$. By a similar argument, one can prove the existence of an infinite chain of indexes $k_0, k_1, \ldots$ as stated in *(ii)* and *(iv)*. Formulæ (27) and (28) immediately imply *(v)*. Finally, in order to prove *(vi)*, it suffices to observe that, first, $*$-intervals can only be units, and, second, that, thanks to (29), a $*$-interval on some interval $[y_{i-1}, y_i]$ for some i implies that y_{i-1} ends a C-interval or y_i starts a C-interval. Thus, if, for the sake of contradiction, $[z, t]$ is a $*$-interval different from the first or the last unit of a C-interval, we have a contradiction with point *(iv)*.

Given the structure built in Lemma 8, we can refer to the m-th $\neg*$-interval of a level C as the m-th tile of that level, and we are therefore interested in connecting the m-th tile of a given level with the m-th tile of the next one. In particular, we say that the tile $[z, t]$ is *connected* to the tile $[z', t']$ if and only if the interval $[t, z']$ is a $Corr$-interval, where $Corr$ is a propositional letter introduced in the following formula:

$$\varphi_{corr} = \quad [G](u \wedge \neg* \rightarrow \langle AO\rangle Corr) \tag{30}$$

$$\wedge\, [G]((u \wedge \neg * \wedge [AO]\neg*) \rightarrow \langle \overline{AO}\rangle Corr) \tag{31}$$

$$\wedge\, [G](Corr \rightarrow [AO]\neg u_e) \tag{32}$$

$$\wedge\, [G](Corr \rightarrow [\overline{AO}]\neg u_b) \tag{33}$$

$$\wedge\, [G](* \rightarrow [\overline{AO}]\neg Corr) \tag{34}$$

$$\wedge\, [G]((u \wedge \langle AO\rangle *) \rightarrow [\overline{AO}]\neg Corr) \tag{35}$$

$$\wedge\, [G](Corr \rightarrow ([DBE]\neg Corr \wedge [DBE]\neg C \wedge [\overline{DBE}]\neg C \wedge \neg C)). \tag{36}$$

Lemma 9. *Let $M, [x, y] \Vdash \varphi_{u_{chain}} \wedge \varphi_{u_{be}} \wedge \varphi_{conf} \wedge \varphi_{corr}$, let $y_0, y_1, y_2, \ldots$ be the infinite sequence of points whose existence is guaranteed by Lemma 6, and let $k_0, k_1, k_2 \ldots$ be the infinite sequence of indexes whose existence is guaranteed by Lemma 8.*

(i) For each $i, j \geq 0$, if the interval $[y_{k_j+i}, y_{k_j+i+1}]$ is a $\neg*$-interval, then the point y_{k_j+i+1} starts a Corr-interval;
(ii) For each $i, j > 0$, if the interval $[y_{k_j+i+1}, y_{k_j+i+2}]$ is a $\neg*$-interval, then the point y_{k_j+i} ends a Corr-interval;
(iii) If $[z, t] \neq [y_i, y_m]$ for each $i, m \geq 0$, then $M, [z, t] \Vdash \neg Corr$, unless $t \leq x$ or $z > y_i$ for each $i \in \mathbb{N}$;
(iv) For each $j > 0$, none of the points y_{k_j}, y_{k_j-1}, and y_{k_j-2} finish any Corr-interval;
(v) The m-th tile of every level is connected to the m-th tile of the next level, and, if the level is not the first one and m is not the index of the last tile of a level, the m-th tile is also connected to the m-th tile of the preceding level;
(vi) Every level has precisely as many tiles as the preceding level plus one.

Proof. Reasoning as in Lemma 7, point *(i)* is an immediate consequence of (30) and (33), and point *(ii)* is an immediate consequence of (31) applied to the interval $[y_{k_j+i}, y_{k_j+i+1}]$. To prove *(iii)*, observe that (32) and (33) mandate that $Corr$-intervals begin and end on points of the type y_i. As far as *(iv)* is concerned, it turns out that neither y_{k_j} or y_{k_j-1} may end a $Corr$-interval because of (34), and y_{k_j-2} may not, either, because of (35). A combinatorial argument, whose cornerstone is (36), proves now *(v)* and *(vi)*. As a matter of fact, $Corr$-intervals that start at a given tile cannot end within the same C-interval or after the next C-interval (thanks to (36)). But then, it is easy to observe that if the m-th tile of a certain C-interval is connected with the l-th tile ($l \neq m$) of the next C-interval one has a contradiction, again, with (36). Finally, thanks to *(iv)*, the last tile of every C-interval cannot be connected to any tile of the previous C-interval, completing the argument.

We complete our construction (pictured in Fig. 6) by forcing that the tiling constraints are respected. Let φ_{color} be the following formula:

$$\varphi_{color} = \bigwedge_{\tau\in\mathcal{T}} [G](\tau \to [AO](Corr \to \langle AO\rangle(\bigvee_{\tau'\in\mathcal{T}, up(\tau)=down(\tau')} \tau'))) \tag{37}$$

$$\wedge \bigwedge_{\tau\in\mathcal{T}} [G]((\tau \wedge [AO]\neg *) \to [AO](\bigvee_{\tau'\in\mathcal{T}, right(\tau)=left(\tau')} \tau')). \tag{38}$$

Lemma 10. *The satisfiability problem for* HS_7 *interpreted over any class of linear frames that contains at least one infinitely ascending order can be reduced to the OTP.*

Proof. Let $\mathcal{T}$ be an instance of the OTP, and let

$$\varphi_{\mathcal{T}} = \varphi_{u_{chain}} \wedge \varphi_{u_{be}} \wedge \varphi_{conf} \wedge \varphi_{corr} \wedge \varphi_{color}.$$

Suppose, first, that $M, [x, y] \Vdash \varphi_{\mathcal{T}}$, where M is a model whose frame belongs to the class of linear frames. By Lemma 8, the point $y = y_0$ starts a first level of three units, containing precisely one tile. By Lemma 9, each successive level contains precisely one more tile than the preceding one, and, by (37) and (38), each m-th tile respects the horizontal constraint with the $m+1$-th tile (if exists), and the vertical constraint with the m-th tile of the successive level. Therefore, M represents a correct tiling of the second octant, and, in summary, the fact that $\varphi_{\mathcal{T}}$ is satisfiable (for a given set of tiles $\mathcal{T}$) implies that the OTP has a solution (for that specific set $\mathcal{T}$). On the contrary, given a solution to the OTP, it is immediate to build a model for $\varphi_{\mathcal{T}}$ by simply placing every tile in its correct position on each level, and suitably filling up the model with the evaluation of $*$, C, and $Corr$. In other words, for a given set $\mathcal{T}$, $\mathcal{T}$ tiles the second octant if and only if $\varphi_{\mathcal{T}}$ is satisfiable. □

Theorem 4. *The satisfiability problem for* HS_7 *interpreted in the class* Fin, Dis, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, *and* Lin, *is undecidable.*

Proof. The class of all finite linear orders is covered by [30]. By Lemma 10, the satisfiability problem for HS_7 in every other case can be reduced to the OTP, and therefore it is undecidable as well. □

6. Experimental results

The results of Section 4 imply that we can implement a relatively efficient finite satisfiability checker for HS_3. In this section, we describe the data structures, the implementation strategies, and the policies we have used; most of the ideas can be also used to implement a satisfiability checker for the case of natural numbers, and, in general, for the discrete case.

Implemented decision procedures for interval temporal logics are not common in the literature. Among the few exceptions, a procedure for the fragment A of HS has been implemented in [16,23]; the former is an experimental implementation not aimed to computational efficiency, and the latter is an attempt to use an automatic tableaux generator, namely MetTeL^2 [43]. The only previous attempt to apply a generic theorem prover to an interval temporal logic can be found in [18], where a tableau-based decision procedure for the fragment D, interpreted over dense linear orders, was developed in LoTREC [24]. Unfortunately, there are no benchmark problems for satisfiability of interval logic formulæ, which makes it difficult to compare different implementations.

6.1. Implementation

We have chosen to develop a *semantic tableau* satisfiability checker for HS_3. Both the tableau and the formula to be checked are represented as rooted decorated trees. A *rooted tree* is a tree $G = (V, E, r)$, where V is a nonempty set, $E \subseteq V \times V$, $|E| = |V| - 1$, and $r \in V$ is its *root*; every element of V is called a *node*. A *rooted decorated tree* [27] is a rooted tree enriched with a function that associates every node with its *decoration*, which can be thought of as the information carried by that node: when we represent formulæ, a decoration is a propositional letter or an operator, while when we represent semantic tableaux, a decoration consists of the information needed to expand the tableau or to close it. In any (rooted decorated) tree, nodes without successors are called *leaves*, and every finite path from the root to a leaf is called a *branch*.

Formulæ are represented as *binary* rooted decorated trees. Each node of the tree is decorated with a code that represents a Boolean or a modal operator or a propositional letter. Formulæ are read and contextually represented as trees; a simple recursive procedure eliminates all implications and pushes all negations in front of propositional letters, obtaining equivalent formulæ in *negated normal form*. Moreover, before checking its satisfiability, our procedure transforms a formula φ into the formula $\varphi \vee \langle L\rangle\varphi \vee \langle I\rangle\varphi$, whose initial satisfiability is checked (see Section 4).

A tableau is represented as a *k-ary* tree in the form of left-child right-sibling. Each node of this tree contains a pointer to the node in the formula tree that represents the sub-formula under analysis, the interval over which it holds, an active/inactive flag, a leaf/internal flag, and the pointers to the left child, the right sibling, and the parent. Domains are represented as totally ordered sets of *floating point* numbers, so that an interval is a pair of floats. This has a very specific purpose: whenever a new point must be added in between a pair of already existing ones, it can be created by simply computing their average; yet, the model remains always finite by construction. Some tableau nodes are leaves during the construction

Table 2
Expansion rules of the semantic tableau-based procedure.

$\dfrac{\psi \vee \tau, [x, y], \mathbb{D}}{\psi, [x, y], \mathbb{D} \mid \tau, [x, y], \mathbb{D}}$		$\dfrac{\psi \wedge \tau, [x, y], \mathbb{D}}{\begin{array}{c}\psi, [x, y], \mathbb{D} \\ \tau, [x, y], \mathbb{D}\end{array}}$
$\dfrac{\langle X\rangle\psi, [x, y], \mathbb{D}}{\psi, [x_1, y_1], \mathbb{D}_1 \mid \ldots \mid \psi, [x_n, y_n], \mathbb{D}_n}$	where $\begin{cases} n = \nu_\Diamond(X, [x, y], \mathbb{D}) \\ \mathbb{D}_i = \mu_\Diamond(i, X, [x, y], \mathbb{D}) \\ [x_j, y_j] = \lambda_\Diamond(j, X, [x, y], \mathbb{D}) \end{cases}$	
$\dfrac{[X]\psi, [x, y], \mathbb{D}}{\begin{array}{c}\psi, [x_1, y_1], \mathbb{D} \\ \psi, [x_2, y_2], \mathbb{D} \\ \cdots \\ \psi, [x_m, y_m], \mathbb{D}\end{array}}$	where $\begin{cases} m = \nu_\Box(X, [x, y], \mathbb{D}) \\ [x_j, y_j] = \lambda_\Box(j, X, [x, y], \mathbb{D}) \end{cases}$	

of the tableau; they (temporally) represent their branch, so that they also store domain information, plus other data that help us choosing the next branch depending on the expansion policy.

In addition to the two trees (a formula tree and a tableau tree – the former is fixed during the satisfiability checking process of a given formula, the latter evolves), there are additional data structures that help us to apply the tableau expansion rules. In particular, the collection of all current leaves is stored in a linked list. Each element of the list points to the tableau node that represents that leaf (and therefore its branch) which may be chosen in the next expansion step. A non-trivial adaptation of a generic tree visit algorithm must be implemented in order to correctly identify, given a tableau node, the set of all and only those leaves that belong to the sub-tree rooted at it. Finally, a dynamic data structure is built (and destroyed) before each expansion step that allows us to examine the branch on which the next to-be-expanded tableau node lies, in order to establish if the branch is closed (because it is contradictory), or it represents a model (in which case the procedure stops and returns that the given formula is satisfiable). Such a structure may be thought of as a hash table that contains the intervals and the formulæ that are (currently) true on them, with an efficient (constant time) lookup method.

The *initial tableau* for a formula φ whose initial satisfiability must be checked is a tableau tree composed of a single tableau node with the following decoration: its formula node pointer has the address of the root of the formula tree that represents φ, its interval is $[0, 1]$, its active flag and its leaf flag are both 1, its domain is $\{0, 1\}$, and all the other pointers are null. Given a leaf of the current tableau (that is, the branch represented by it), the following two operations are performed: *(i)* branch closing checking and branch model checking, and *(ii)* choosing the next to-be-expanded tableau node. Assuming that the current branch B is not closed and is not a model, the next to-be-expanded tableau node is chosen according to the following policy: it is the active node on B that is closest to the root. Expansion rules are described in Table 2. Boolean rules are standard, while the rules for modal operators are designed as follows. Let $\mathfrak{I}$ be the set of all intervals in any finite domain. As for the existential cases, we define a function

$$\nu_\Diamond : \{I, L, \overline{L}\} \times \mathfrak{I} \times \mathsf{Fin} \to \mathbb{N},$$

with the following meaning. Given an interval $[x, y]$ and a domain $\mathbb{D}$, $\nu_\Diamond(X, [x, y], \mathbb{D})$ returns the number of already existing different intervals in the relation R_X with $[x, y]$ plus the number of new intervals that should be created in the relation R_X in order to explore all qualitatively distinct possibilities. For example, $\nu_\Diamond(L, [0, 1], \{0, 1, 2\}) = 5$: indeed, if $\langle L\rangle\psi$ holds on $[0, 1]$ and the current domain is $\{0, 1, 2\}$, then ψ may hold on some interval $[1.25, 1.75]$, $[1.5, 2]$, $[2, 2.5]$, $[3, 4]$, or $[1.5, 2.5]$. Notice that, for example, adding $[1.25, 1.75]$ is necessary as it represents a new interval completely between existing points; something similar happens for $[3, 4]$, because $\langle L\rangle\psi$ holding on $[0, 1]$ may be satisfied by a new interval starting after the point 2. Similarly, as for the universal case, we define a function $\nu_\Box$, with the same parameters, that returns only the number of already existing different intervals in the relation R_X with $[x, y]$ (indeed, in order to expand a universal modality, no new intervals are created). Moreover, expanding an existential modality may require that some branch have a new (bigger) domain (because, as we have seen above, existential modalities may require new points); the i-th domain (in no particular order) is returned by the function

$$\mu_\Diamond : \mathbb{N} \times \{I, L, \overline{L}\} \times \mathfrak{I} \times \mathsf{Fin} \to \mathsf{Fin}.$$

Finally, while expanding any existential modality, every new node in a branch is decorated with an interval that belongs to the (possibly new) domain. The j-th interval (in no particular order) of the expansion is returned by the function

$$\lambda_\Diamond : \mathbb{N} \times \{I, L, \overline{L}\} \times \mathfrak{I} \times \mathsf{Fin} \to \mathfrak{I}.$$

Following up with the previous example, $\mu_\Diamond(1, L, [0, 1], \{0, 1, 2\}) = \{0, 1, 1.25,\ 1.75, 2\}$ and $\lambda_\Diamond(1, L, [0, 1], \{0, 1, 2\}) = [1.25, 1.75]$. Because we are restricting ourselves to initial satisfiability, these functions never return domains or intervals with new points between 0 and 1, nor smaller than 0. In this way for each existential operator $\langle X\rangle\psi$ we have an existential disjunctive rule that creates enough branches to search for every possible location of ψ. Universal operators are easier to

treat: given the current interval and the current domain, we use the function $\lambda_{\Box}$, analogous to the previous one, to obtain all intervals on which a formula must be applied, and since universal operators do not add points to the domain, the latter, as we have already observed, does not change.

Given the node to be expanded, the correct rule is chosen for its expansion (clearly, precisely one rule can be applied on it); the result of such a step is applied to *all leaves in the sub-tree* rooted at the chosen node. For efficiency reasons, decorations are never duplicated. A given branch B, identified by its leaf, is *closed* if one of the following two conditions hold: a propositional contradiction is found on it, that is, there exist two nodes on B such that their decorations show $p, [x, y]$ and $\neg p, [x, y]$, respectively, or the cardinality of the domain in the decoration of its leaf exceeds the bound (see Lemma 4). When a given leaf is selected for the next expansion step, the branch that contains it is checked for contradictions. If a contradiction is found, the leaf is simply eliminated from the list of leaves; if all leaves are eliminated, then the formula is found unsatisfiable. Otherwise, the node is expanded and the protocol for managing active/inactive flags, after every expansion, is as follows: nodes are inactivated after being expanded, but in case of universal nodes, they are copied at the end of the branch with the active flag at 1. Therefore, the current branch B is a model if and only if, for every active node on B, that node is universal, and, if $\mathbb{D}$ is in the decoration of its leaf, then, for every $[X]\psi, [x, y]$ in the decoration of some node of B there exists a node with $\psi, [x_j, y_j]$ in B, for every $1 \leq j \leq \nu_{\Box}(X, [x, y], \mathbb{D})$.

The soundness and the completeness of our procedure are shown in the following result.

Theorem 5. *A formula φ of HS_3 is finitely satisfiable if and only if the tableau-based method, with the rules in Table 2, returns 'Satisfiable'.*

Proof. Let us introduce first the following notion. Consider a node n on a tableau for a formula φ, and let $S(n)$ be the set of all decorations on nodes between n and the root. We denote with $\mathbb{D}(n)$ the domain in the decoration of n. We say that $S(n)$ is *satisfied on an extension of* $\mathbb{D}(n)$ if there exists a model M based on some extension of $\mathbb{D}(n)$ such that, for each $(\psi, [x, y]) \in S(n)$ it is the case that $M, [x, y] \Vdash \psi$.

We now show that for every finitely satisfiable formula of HS_3 the presented method terminates and returns 'Satisfiable', that is, contra-positively, whenever the procedure closes all branches, the formula is not finitely satisfiable. To do so, we prove a stronger claim: for any node n at height h on a tableau for φ, if every branch that contains n is closed, then $S(n)$ is not satisfied on any extension $\mathbb{D}'$ of $\mathbb{D}(n)$ such that $|\mathbb{D}'| \leq LIM$ (where LIM is the maximum number of points on a domain as shown in Lemma 4). Notice that, when n is the root, this is to say that φ is not finitely satisfiable. We proceed now to prove the claim by induction. If $h = 0$, either the only branch that contains n also contains two nodes with decorations $(p, [x, y])$ and $(\neg p, [x, y])$, and therefore $S(n)$ is simply not satisfiable, or the number of points ever named in the decorations of its nodes is more than LIM, for which $S(n)$ can never be satisfied on any extension of $\mathbb{D}(n)$. If $h > 0$, then n has been expanded by some rule, some nodes $n_1, n_2, \ldots$ exist that are descendants of n, and the inductive hypothesis applies to all of them. If the rule that has been applied is Boolean, then the claim follows immediately. If it is the universal rule, then suppose that $([X]\psi, [x, y])$ is in the decoration of n. Every branch that contains n also contains all nodes that are the result of its expansion, and, in particular, some node n' with decoration $(\psi, [z, t])$ for some interval $[z, t]$ such that $[x, y]R_X[z, t]$. If $S(n)$ were satisfiable on some extension of $\mathbb{D}(n)$, then, in particular, $S(n) \cup \{(\psi, [z, t])\} = S(n')$ would be too, but this is in contradiction with the inductive hypothesis. Finally, if it is the existential rule, then suppose that $(\langle X\rangle\psi, [x, y])$ is in the decoration of n. If $S(n)$ were satisfiable on some extension of $\mathbb{D}(n)$, then there would be a model whose domain extends $\mathbb{D}(n)$ such that it satisfies $\langle X\rangle\psi$ on $[x, y]$ and ψ on some $[z, t]$ such that $[x, y]R_X[z, t]$. By construction, there must be some successor n' of n that contains the decoration $(\psi, [z, t])$, independently of z, t being already in $\mathbb{D}(n)$. This means that $S(n')$ would be satisfiable on some extension of $\mathbb{D}(n')$, which is in contradiction with the inductive hypothesis.

Now, we must argue that our method is also sound, that is, for every formula φ of HS_3 for which it returns 'Satisfiable', there exists a model M such that $M, [0, 1] \Vdash \varphi$. Consider a branch B such that it is not contradictory, all its active nodes are universal, and every node with universal decoration has been already expanded on every possible interval of the domain $\mathbb{D}$ of the branch. Now, let M be a model based on $\mathbb{D}$, and whose valuation function is defined as follows: for each interval $[x, y]$ and each propositional letter p, $[x, y] \in V(p)$ if and only if $(p, [x, y])$ decorates some node on B. We want to prove, by structural induction, that, for each node n in B with decoration $(\psi, [x, y])$, $M, [x, y] \Vdash \psi$. If ψ is a propositional letter or its negation, we have the result immediately. If ψ is a composite formula, two cases arise: either it is a universal formula, or it is not. In the latter case, the fact that B is not closed implies that n has been expanded, and such expansion has been applied to all branches that contain n: if ψ is a conjunction, then both conjuncts have been included as decorations in nodes of B, if it is a disjunction then at least one disjunct has been included as decoration in some node of B, and, if it is $\psi = \langle X\rangle\xi$, then at least one node in B must be decorated with $(\xi, [z, t])$, for some $[z, t]$ such that $[x, y]R_X[z, t]$. In all cases, the inductive hypothesis applies, and therefore M must satisfy ψ on $[x, y]$. In the former case, if ψ is a universal formula, i.e. $\psi = [X]\xi$, since B cannot be further extended, it must be the case that a node n' with decoration $(\xi, [z, t])$ occurs in B for each $[z, t]$ such that $z, t \in \mathbb{D}$ and that $[x, y]R_X[z, t]$, and again, the inductive hypothesis applies. □

6.2. Benchmark and experimental results

Our procedure is programmed object-oriented in C++ standard language with *threads* capabilities. Threads are run virtually in parallel, and carry a specific policy for choosing the next leaf to be examined. Each policy is *fair*, that is, every

Table 3
A benchmark for (finitely) satisfiable formulæ of HS_3.

Finitely satisfiable	
name	formula
φ_1	$p_0 \wedge [I]\neg p_2 \wedge [\overline{L}]\neg p_2 \wedge \langle L\rangle p_1 \wedge [L](p_1 \rightarrow ([I]\neg p_3 \wedge [L]\neg p_3 \wedge \langle \overline{L}\rangle p_2))$
φ_3	$p_0 \wedge [I]\neg p_2 \wedge [\overline{L}]\neg p_2 \wedge \langle L\rangle p_1 \wedge [L](p_1 \rightarrow ([I]\neg p_3 \wedge [L]\neg p_3 \wedge \langle \overline{L}\rangle \varphi_1^{+2}))$
φ_5	$p_0 \wedge [I]\neg p_2 \wedge [\overline{L}]\neg p_2 \wedge \langle L\rangle p_1 \wedge [L](p_1 \rightarrow ([I]\neg p_3 \wedge [L]\neg p_3 \wedge \langle \overline{L}\rangle \varphi_3^{+2}))$
...	...

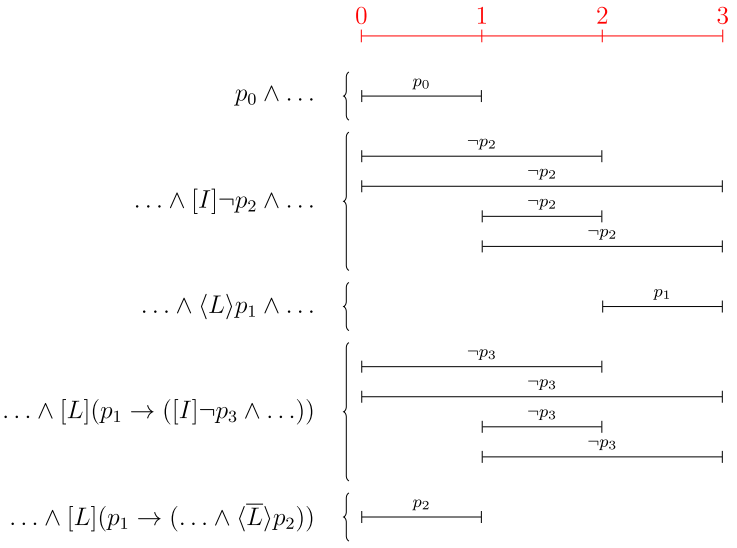


Fig. 7. Example of a possible model for φ_1 from Table 3.

branch is eventually examined; the advantage of using different policies is the improved execution time, especially for satisfiable formulæ. We took into account two key aspects: domain cardinality and branch sparseness. The *sparseness* degree of a branch allows us to estimate how many distinct intervals in the domain are actually used; to compute such an estimation, we calculate the average of positive propositional letters assigned to each interval, and define the sparseness of the branch as the variance of the distribution associated to assigning positive propositional letters to intervals. We implemented the following policies: *(i)* branches with smaller domain and less sparse first (SBF); *(ii)* branches with longer domain and more sparse first (LBF); *(iii)* branches in Last-In-First-Out order (LIFO) – that is, the tableau tree is explored depth-first. All experiments have been carried out on an Intel(R) Core(TM) i7-6700HQ, with a clock of 2.60 GHz, four cores, and 16 GB RAM.

Robustness and scalability. In order to test the robustness and the scalability of our implementation, we designed the sequence of finitely satisfiable formulæ shown in Table 3, which are systematically generated for $k = 1, 3, 5, \ldots$, so that their length can be put in relation with the time that our program takes to establish its satisfiability. Formulæ are generated inductively: given φ_k, we obtain φ_k^{+2} by replacing each propositional letter p_i with p_{i+2}, and, from φ_k^{+2}, we generate φ_{k+2}; each φ_k is finitely satisfiable and it has only few different models. More importantly, the formulæ φ_k grow in length and in modal depth; following [9], length, modal depth, and number of different models are three of the most important indicators to measure the experimental difficulty of proving the satisfiability of a formula. The results of our scalability experiment are shown in Fig. 8.

For the sake of clarity, Fig. 7 shows a four points model of φ_1 (from Table 3), as returned by our algorithm. In Fig. 7, on the left-hand side, we consider a sub-formula of φ_1 and, correspondingly, on the right-hand side, we show how the considered sub-formula is, in fact, satisfied in the model. Observe that, in line with classical tableau-based algorithms, only necessary requests are shown: for example, there is no obligatory truth value for p_1 on $[0, 1]$ because φ_1 does not require it. Moreover, trivially satisfied sub-formulas are not shown: for instance, on $[0, 1]$, the sub-formula $[\overline{L}]\neg p_2$ is satisfied by the fact that 0 is the first point of the model.

Semi-randomized test. In order to test our implementation in a more realistic scenario, we need formulæ whose satisfiability status is not fixed. Purely random formulæ are not necessarily challenging for a satisfiability tester: as a matter of fact, random formulæ tend to be satisfiable, often admitting several different models. Moreover, testing the scalability of the system using only unsatisfiable formulæ of increasing length is not appropriate, as the effort needed to prove that a

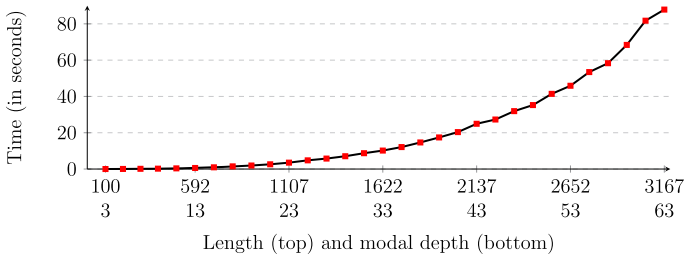


Fig. 8. Elapsed time for satisfiable formulæ (by length and modal depth).

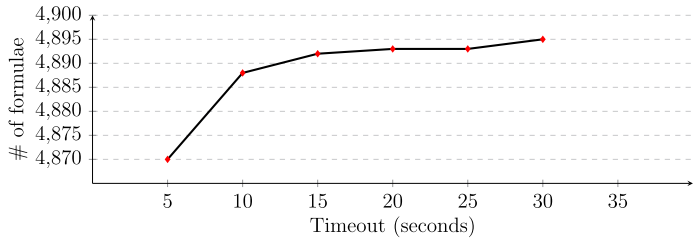


Fig. 9. Timeouts versus number of finalized formulæ.

formula is unsatisfiable depends on many aspects besides its length, such as, for example, the depth (in the tableau) at which a contradiction is found. Therefore, we proceeded as follows.

First, we designed a simple technique to systematically generate scalable unsatisfiable formulæ. To this end, we listed a certain number of propositional tautologies, such as

$$(p \wedge (p \rightarrow q)) \rightarrow q,$$
$$(p \wedge q) \rightarrow p,$$
$$(p \rightarrow (p \vee q)),$$

as well as a certain number of modal tautologies of HS_3, such as

$$\langle L\rangle\langle L\rangle p \rightarrow \langle L\rangle p, \langle L\rangle p \rightarrow [\overline{L}]\langle L\rangle p.$$

Then, we systematically applied several cycles of *uniform substitution* to our initial group of tautologies, obtaining longer ones. Once we were satisfied of the average length of the resulting formulæ (around 150 symbols each), we negated them, obtaining unsatisfiable formulæ.

Second, we designed a simple method to generate randomized formulæ of controllable length, over an alphabet $\mathcal{AP}$ of propositional letters, by employing a very intuitive schema that recursively produces a formula tree: given the current height h, we randomly choose a Boolean or a modal operator (with probability directly proportional to h), or a propositional letter in the alphabet (with probability inversely proportional to h), and perform one (or two, depending on the case) recursive call(s) with height $h-1$. It may be argued that such procedure does not guarantee that the obtained formula has a *specific* length; it does, however, guarantee that all formulæ generated with it are, *on average*, of the same length, and we can suitably modulate the initial value of h to obtain formulæ that tend to be of the desired length. Modulating the proportion between h and $|\mathcal{AP}|$ gives us some control on the probability that the generated formula is indeed satisfiable (following the same principles as in [25]).

Finally, we created a database of 5000 formulæ, of comparable length, that contains formulæ from both groups, on a proportion 40% (the former, certainly unsatisfiable) and 60% (the latter, probably satisfiable).

We tested our procedure against this problem in order to establish which fraction of the entire group could be solved under different timeout settings, from 5 to 30 seconds. The results of these experiments are depicted in Fig. 9. As it can be seen, a timeout of 15 seconds was enough to establish the satisfiability of over the 97% of our tested formulæ.

7. Conclusions

In this paper we studied two previously unknown variants of Halpern and Shoham's logic (HS), inspired by Golumbic and Shamir's interval algebras, which generalize the classical Allen's Interval Algebra with coarser interval relations. While the finer HS_7 is still generally undecidable, the coarser HS_3 becomes PSPACE-complete in the finite case, the case of the natural numbers, and the integers, and PSPACE-hard in the other cases. A summary of the known results on coarser interval temporal logics can be found in Table 4. Undecidability of HS in the finite case can be proven as in [28], although the

Table 4
A summary of the results of this paper.

Class	HS	HS_7	HS_3
Fin, $\mathbb{N}$, $\mathbb{Z}$, Dis	Undec. [28]	Undec. [30]	PSPACE-complete
Lin, $\mathbb{Q}$	Undec. [28]	Undec.	PSPACE-hard

authors do not mention this particular case; many of the stricter undecidability results for fragments of HS such as, for example, those in [12], actually imply that HS is undecidable in this case as well. Fragments of HS have been extensively studied in the recent literature; unfortunately, there is no natural comparison between coarser fragments (such as HS_3 and HS_7) and syntactical fragments of HS. The latter have been systematically analyzed both from the relative expressive power [1] and from the computational complexity point of view [14,15].

There are a number of open problems and interesting research directions in this topic. Besides the natural interest in completing the picture of the decidability and the complexity of the satisfiability problem for coarser interval temporal logics, there are obvious questions that arise from our results. For example, it has been proved that in the finite/discrete case the fragment $\mathsf{L}\overline{\mathsf{L}}$ is NP-complete [14], which means that there may exist natural and useful logics expressively *in between* $\mathsf{L}\overline{\mathsf{L}}$ and HS_3. Moreover, there exists a recently emerged interest in interval temporal logic under sub-propositional restrictions [20,7,19], and it makes sense to explore a logic such as HS_3 under similar restrictions.

Acknowledgements

This work was partially supported by the project TIN15-70266-C2-P-1 (E. Muñoz-Velasco), founded by the Spanish Ministry of Science, the PhD grant FPU15/05883 (M. Pelegrín), founded by Spanish Ministry of Education, and by the project Formal Methods for Verification and Synthesis of Discrete and Hybrid Systems (P. Sala and G. Sciavicco), founded by the Italian INDAM GNCS.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.artint.2018.09.001.

References

[1] L. Aceto, D. Della Monica, V. Goranko, A. Ingólfsdóttir, A. Montanari, G. Sciavicco, A complete classification of the expressiveness of interval logics of Allen's relations: the general and the dense cases, Acta Inform. 53 (3) (2016) 207–246.
[2] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1983) 832–843.
[3] R. Alluhaibi, Simple interval temporal logic for natural language assertion descriptions, in: Proc. of the 11th International Conference on Computational Semantics, IWCS, 2015, pp. 283–293.
[4] A. Artale, D. Bresolin, A. Montanari, V. Ryzhikov, G. Sciavicco, DL-Lite and interval temporal logics: a marriage proposal, in: Proc. of the 21st European Conference of Artificial Intelligence, ECAI, 2014, pp. 957–958.
[5] A. Artale, E. Franconi, A temporal description logic for reasoning about actions and plans, J. Artif. Intell. Reason. 9 (1998) 463–506.
[6] A. Artale, R. Kontchakov, V. Ryzhikov, M. Zakharyaschev, A cookbook for temporal conceptual data modelling with description logics, ACM Trans. Comput. Log. 15 (3) (2014) 25:1–25:50.
[7] A. Artale, R. Kontchakov, V. Ryzhikov, M. Zakharyaschev, Tractable interval temporal propositional and description logics, in: Proc. of the 29th AAAI Conference on Artificial Intelligence, AAAI, 2015, pp. 1417–1423.
[8] R.A. Baeza-Yates, Challenges in the interaction of information retrieval and natural language processing, in: Proc. of the 5th International on Computational Linguistics and Intelligent Text Processing, CICLing, 2004, pp. 445–456.
[9] P. Balsiger, A. Heuerding, S. Schwendimann, A benchmark method for the propositional modal logics K, KT, S4, J. Autom. Reason. 24 (3) (2000) 297–317.
[10] C. Bettini, Time-dependent concepts: representation and reasoning using temporal description logics, Data Knowl. Eng. 22 (1) (1997) 1–38.
[11] E. Börger, E. Grädel, Y. Gurevich, The Classical Decision Problem, Perspectives of Mathematical Logic, Springer, 1997.
[12] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, G. Sciavicco, Decidable and undecidable fragments of Halpern and Shoham's interval temporal logic: towards a complete classification, in: Proc. of the 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR, in: Lect. Notes Comput. Sci., vol. 5330, Springer, 2008, pp. 590–604.
[13] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, G. Sciavicco, The dark side of interval temporal logic: marking the undecidability border, Ann. Math. Artif. Intell. 71 (1–3) (2014) 41–83.
[14] D. Bresolin, D. Della Monica, A. Montanari, P. Sala, G. Sciavicco, Interval temporal logics over strongly discrete linear orders: expressiveness and complexity, Theor. Comput. Sci. 560 (2014) 269–291.
[15] D. Bresolin, D. Della Monica, A. Montanari, P. Sala, G. Sciavicco, On the complexity of fragments of the modal logic of Allen's relations over dense structures, in: Proc. of the 9th International Conference on Language and Automata Theory and Applications, LATA, in: Lect. Notes Comput. Sci., vol. 8977, Springer, 2015, pp. 511–523.
[16] D. Bresolin, D. Della Monica, A. Montanari, G. Sciavicco, A tableau system for right propositional neighborhood logic over finite linear orders: an implementation, in: Proc. of the 22nd International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX, in: Lect. Notes Comput. Sci., vol. 8123, 2013, pp. 74–80.
[17] D. Bresolin, D. Della Monica, A. Montanari, G. Sciavicco, The light side of interval temporal logic: the Bernays–Schönfinkel fragment of CDT, Ann. Math. Artif. Intell. 71 (1–3) (2014) 11–39.
[18] D. Bresolin, V. Goranko, A. Montanari, P. Sala, Tableaux for logics of subinterval structures over dense orderings, J. Log. Comput. 20 (1) (2010) 133–166.
[19] D. Bresolin, A. Kurucz, E. Muñoz-Velasco, V. Ryzhikov, G. Sciavicco, M. Zakharyaschev, Horn fragments of the Halpern–Shoham interval temporal logic, ACM Trans. Comput. Log. 18 (3) (2017) 22:1–22:39.

[20] D. Bresolin, E. Muñoz-Velasco, G. Sciavicco, Sub-propositional fragments of the interval temporal logic of Allen's relations, in: Proc. of the 14th European Conference on Logics in Artificial Intelligence, JELIA, in: Lect. Notes Comput. Sci., vol. 8761, Springer, 2014, pp. 122–136.
[21] Z. Chaochen, M.R. Hansen, Duration Calculus: A Formal Approach to Real-Time Systems, EATCS: Monographs in Theoretical Computer Science, Springer, 2004.
[22] J. Chomicki, D. Toman, Temporal logic in information systems, in: Logics for Databases and Information Systems, 1998, pp. 31–70.
[23] D. Della Monica, A. Montanari, G. Sciavicco, D. Tishkovsky, First steps towards automated synthesis of tableau systems for interval temporal logics, in: Proc. of the 5th International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, COMPUTATION TOOLS, 2014, pp. 32–37.
[24] L. Fariñas del Cerro, D. Fauthoux, O. Gasquet, A. Herzig, D. Longin, F. Massacci, Lotrec: the generic tableau prover for modal and description logics, in: Proc. of the 1st International Joint Conference on Automated Reasoning, IJCAR, 2001, pp. 453–458.
[25] I.P. Gent, T. Walsh, The SAT phase transition, in: Proc. of the 11th European Conference on Artificial Intelligence, ECAI, 1994, pp. 105–109.
[26] M.C. Golumbic, R. Shamir, Complexity and algorithms for reasoning about time: a graph-theoretic approach, J. ACM 40 (5) (1993) 1108–1133.
[27] V. Goranko, A. Montanari, G. Sciavicco, Propositional interval neighborhood temporal logics, J. Univers. Comput. Sci. 9 (9) (2003) 1137–1167.
[28] J. Halpern, Y. Shoham, A propositional modal logic of time intervals, J. ACM 38 (4) (1991) 935–962.
[29] S. Klarman, Practical querying of temporal data via OWL 2 QL and SQL:2011, in: Proc. of the 19th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR, in: EPiC Series, vol. 26, Center for Artificial Intelligence Research, 2014, pp. 52–61.
[30] J. Marcinkowski, J. Michaliszyn, The undecidability of the logic of subintervals, Fundam. Inform. 131 (2) (2014) 217–240.
[31] M. Marx, M. Reynolds, Undecidability of compass logic, J. Log. Comput. 9 (6) (1999) 897–914.
[32] A. Molinari, A. Montanari, A. Murano, G. Perelli, A. Peron, Checking interval properties of computations, Acta Inform. 53 (6–8) (2016) 587–619.
[33] A. Montanari, I. Pratt-Hartmann, P. Sala, Decidability of the logics of the reflexive sub-interval and super-interval relations over finite linear orders, in: Proc. of the 17th International Symposium on Temporal Representation and Reasoning, TIME, IEEE Computer Society, 2010, pp. 27–34.
[34] A. Montanari, G. Puppis, P. Sala, Maximal decidable fragments of Halpern and Shoham's modal logic of intervals, in: Proc. of the 37th International Colloquium on Automata, Languages and Programming – Part II, ICALP, in: Lect. Notes Comput. Sci., vol. 6199, Springer, 2010, pp. 345–356.
[35] A. Montanari, G. Puppis, P. Sala, A decidable weakening of compass logic based on cone-shaped cardinal directions, Log. Methods Comput. Sci. 11 (4) (2015).
[36] A. Montanari, G. Puppis, P. Sala, G. Sciavicco, Decidability of the interval temporal logic $A\bar{B}\bar{B}$ on natural numbers, in: Proc. of the 27th Symposium on Theoretical Aspects of Computer Science, STACS, Inria Nancy Grand Est & Loria, 2010, pp. 597–608.
[37] A. Montanari, G. Sciavicco, N. Vitacolonna, Decidability of interval temporal logics over split-frames via granularity, in: Proc. of the 8th European Conference on Logics in Artificial Intelligence, JELIA, in: Lect. Notes Artif. Intell., vol. 2424, Springer, 2002, pp. 259–270.
[38] B. Moszkowski, Reasoning About Digital Circuits, PhD thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.
[39] C.M. Papadimitriou, Computational Complexity, Addison-Wesley Longman Publishing Co., Inc., 1994.
[40] I. Pratt-Hartmann, Temporal prepositions and their logic, Artif. Intell. 166 (1–2) (2005) 1–36.
[41] A. Schmiedel, Temporal terminological logic, in: Proc. of the 8th National Conference on Artificial Intelligence, AAAI, AAAI Press, 1990, pp. 640–645.
[42] L.J. Stockmeyer, A.R. Meyer, Word problems requiring exponential time (Preliminary Report), in: Proc. of the 5th Annual ACM Symposium on Theory of Computing, STOC, ACM, 1973, pp. 1–9.
[43] D. Tishkovsky, R.A. Schmidt, M. Khodadadi, The tableau prover generator MetTeL^2, in: Proc. of the 13th European Conference on Logics in Artificial Intelligence, JELIA, 2012, pp. 492–495.
[44] Y. Venema, Expressiveness and completeness of an interval tense logic, Notre Dame J. Form. Log. 31 (4) (1990) 529–547.
[45] F. Zemke, What's new in SQL:2011, SIGMOD Rec. 41 (1) (2012) 67–73.