

Core Fragments of Compass Logic

Pietro Casavecchia ✉ 

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

Guido Sciavicco ✉ 

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

Leonardo Serrentino ✉ 

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

Abstract

Compass Logic is arguably the most intuitive modal spatial logic designed for a point-based ontology, featuring exactly one unary modal operator for each cardinal direction, and interpreted over bi-dimensional structures; it can also be seen as a very natural bi-temporal logic, allowing independent reasoning in terms of future and past in two orthogonal temporal dimensions. As such, Compass Logic finds applications in several areas of modern artificial intelligence, from spatial reasoning to temporal databases. Its introduction is due to Marx and Reynolds in 1999 (inspired by previous work of Venema, 1990), when the authors proved that its satisfiability problem is undecidable regardless of the properties of the underlying structure. More recently, the idea of exploring sub-propositional fragments of unary modal logic emerged, and was applied to point-based and interval-based temporal logic. In the same spirit, in this paper we explore sub-propositional fragments of Compass Logic in the finite/discrete case, and we prove that while its satisfiability problem remains undecidable in very weak conditions, such as for the core fragment, characterized by having only formulas that are conjunctions of binary Horn clauses, it becomes surprisingly decidable for the same core fragment devoid of existential quantifiers.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics; Theory of computation → Proof theory

Keywords and phrases spatial logic, bi-temporal logic, Horn fragment, core fragment, undecidability, decidability

Digital Object Identifier 10.4230/OASICS.TIME.2026.2

Related Version *Full Version:* <https://drive.google.com/file/d/10Za315mjSRp51SrK621EVQfEj8PaK9Q/view>

Funding This research was partially funded by INdAM – GNCS Project *Semantic neuro-symbolic contract for the rigorous reachability and CPS certifications* (CUP: E53C25002010001); Guido Sciavicco is a member of INdAM.

1 Introduction

Reasoning about time in terms of future and past, as well as about space in terms of directional relations is a natural idea.

Several qualitative temporal and spatial calculi have been developed for formalizing and reasoning about direction or ordering relations. At the algebraic level, these include the Point Calculus [19], which defines three ordering relations for points in a single-dimension Euclidean space, Allen’s Interval Calculus [2], with thirteen relations between intervals on a linear order, the Cardinal Direction Calculus [14], extending Point Calculus in two dimensions, and the Rectangle Algebra [6], extending, in turn, Allen’s algebra to the two dimensional case. Similarly, at the logical level the main separation among different proposals corresponds to points versus extended areas (in this case, intervals or rectangles) as atomic objects. The most important contributions generally come from the realm of propositional modal logic, and, on



© Pietro Casavecchia, Guido Sciavicco, and Leonardo Serrentino;
licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Temporal Representation and Reasoning (TIME 2026).

Editors: AndreA Orlandini and Sophie Pinchinat; Article No. 2; pp. 2:1–2:16

OpenAccess Series in Informatics



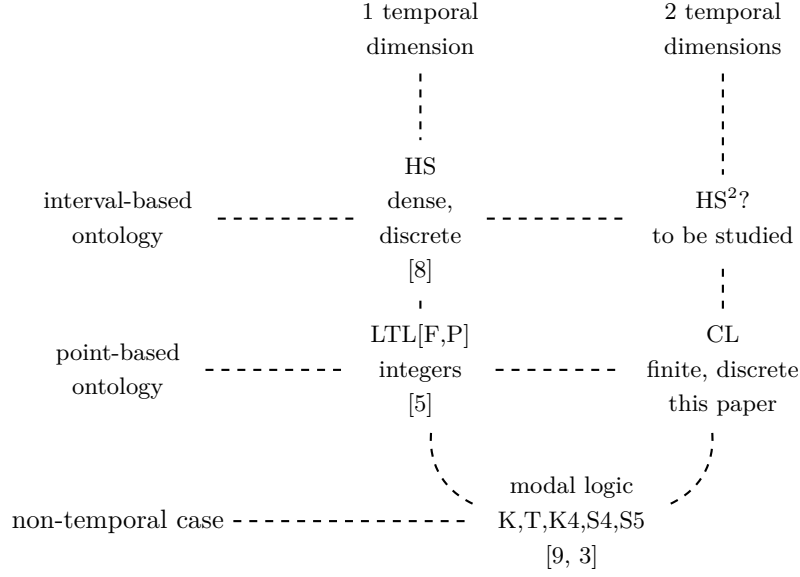
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

a single temporal dimension, include Linear Temporal Logic with Future and Past (LTL[F,P], often presented as fragment of LTL, the Linear Temporal Logic with Since and Until) for the point-based case, and Halpern and Shoham's Modal Logic of Allen's Relations (HS) [12] for the interval-based case, and, in two temporal dimensions, Compass Logic (CL) [16], that is, the logical counterpart of Cardinal Direction Calculus for the point-based case. The immediate logical counterpart to Rectangle Algebra is the full product (in the sense of [11]) of HS with itself, encompassing 169 relations, and probably not very natural; the fragment of the Logic of Topological Relations (L_{RCC8}) [15] restricted to axes-parallel rectangles can also be mentioned in this context. Many other temporal and spatial logical frameworks exist, in fact too many to be reviewed here; however, for the purposes of this paper, we focus on logical languages inspired by, or that can assimilated to bi-temporal relations.

Sub-propositional fragments of modal logic are obtained by limiting the applicability of Boolean operators in formulas, in the spirit of Horn propositional logic. By focusing on binary clauses only, one obtains the so-called Krom fragments; binary Horn fragments are known as core fragments. In general, Horn and core fragments disallow the usual existential/universal duality, effectively generating different languages, with and without either of them, that can be partially ordered by expressive power. It is natural to ask how temporal and bi-temporal logics behave, computationally speaking, when restricted to sub-propositional fragments.

This line of investigation has been pursued in the case of Modal Logic interpreted in the classes of general (K), transitive (K4), reflexive (T), and transitive and reflexive frames (S4) [9], and revealed that the satisfiability problem reacts positively in several combinations of restrictions and class of frames: in all mentioned cases the satisfiability problem, which is PSPACE in the case of the full language, remains so in the Horn fragment, becomes *P*-complete in the case of the Horn fragment without diamonds, is between *NLOGSPACE* and *P* in the case of core fragment without diamonds for in all cases except K, which is *NLOGSPACE*-complete; the case of Horn S5 had already been mentioned in [3], and proved to be *NP*-complete. As for the temporal case, the sub-propositional fragments of LTL[F,P], with and without an additional universal operator, have been studied in [5], over the integers: as it happens, the satisfiability problem, which is *NP* for the full language, reduces to *P*-complete in the case of Horn without diamonds, and to *NLOGSPACE*-complete in the case of core without diamonds. This same approach in the case of linear temporal logic of intervals was attempted in [8] by tackling sub-propositional fragments of HS, whose satisfiability problem is generally undecidable, and revealed a more involuted situation. Indeed, HS interpreted on irreflexive frames (the usual interpretation) is still undecidable (*i*) under the core restriction even without boxes, and (*ii*) the Horn restriction even without diamonds, in the discrete case; it is *PSPACE*-hard under the core restriction, even without boxes, in the discrete case; it is *P* (resp., *P*-complete) under the core (resp., Horn) restriction without boxes. Moreover, HS interpreted on reflexive frames (a less intuitive and common interpretation) is undecidable under the core restriction even without boxes; *PSPACE*-hard under the Horn restriction, and in *P* (resp., *P*-complete) under the core (resp., Horn) restriction even without diamonds. Finally, both the reflexive and irreflexive cases still present several open problems.

In light of the above, stepping from one to two orthogonal temporal dimensions seems very natural; the situation can be graphically represented as in Fig. 1. Compass Logic, originally presented in [16] as a logic to reason about spatial situations (although Venema suggested the idea of Compass Logic and its notation in [18]), can be seen as the full product of LTL[F,P] with itself, allowing independent reasoning on both directions. Satisfiability in CL is undecidable, regardless of the properties of the underlying frame, and the quest for decidable and well-behaved fragments includes two relevant contributions. One of them is Cone Logic [17], which is a decidable weakening of CL obtained by enforcing the restriction



■ **Figure 1** A representation of the current status of research on sub-propositional temporal and bi-temporal logics. Bi-temporal logics can also be seen as spatial logics: in this regard, other prominent frameworks could show computational behaviour improvements under the Horn and core restrictions.

that quantifications along one axis can be used only after a similar quantification along the other axis; in this way, the modalities can be seen as north-east, north-west, south-east, and south-west, and the expressive power of the resulting language lowers to the point that its satisfiability problem becomes decidable. The other one is in fact the collection of all results concerning decidable fragments of HS [1, 7], which can be geometrically interpreted over two dimensions; such a geometric interpretation gives rise to (somewhat unnatural) operators that can be defined within Compass Logic, effectively converting any fragment of the former in a fragment of the latter. In both cases, however, the bi-temporal (and sometimes even the spatial) nature of the logic is lost. Finally, CL itself can be seen as a fragment of the bi-temporal logic proposed in [10], the fragments of the former, even those studied here, can be seen as fragments of the latter. Here, we prove that the satisfiability problem for CL, which is undecidable in the Boolean case, remains so for its core fragment with both universal and existential operators, in the finite (i.e., *has a formula φ a finite model?*), the infinite discrete (*has φ a model with each axes isomorphic to the integers?*), and the general discrete case (*has φ a discrete model?*), assuming, as usual, irreflexivity of the frames (i.e., the relation $<$ is strict).

All proofs can be found in the extended version of this paper.

2 Compass Logic

Let $\mathcal{P}$ be a (possibly infinite, but countable) set of *proposition letters* (or, simply, *propositions*).

► **Definition 1.** *The well-formed formulas of (propositional) Compass Logic (CL) are generated by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi,$$

where $p \in \mathcal{P}$ and $\Diamond \in \{\Diamond, \Diamond, \Diamond, \Diamond\}$. The size of φ , denoted by $|\varphi|$, is the number of its symbols.

Intuitively, formulas of CL will be interpreted on bi-dimensional Kripke structures, and $\diamondleft$ (resp., $\diamondright, \diamonddown, \diamondup$) will be interpreted as *exists to the left* (resp., *right, down, up*). The standard abbreviations apply: $\top, \vee$, and $\rightarrow$ respectively for true, Boolean disjunction, and Boolean implication, and one box operator for each diamond, defined as $\Box p \equiv \neg \diamond \neg p$, for $\Box \in \{\Boxleft, \Boxright, \Boxdown, \Boxup\}$. In the following, we use $|\varphi|$ to denote the length of φ , that is, the number of its symbols.

► **Definition 2.** Let $\mathbb{D} = \langle D, < \rangle$ be a linearly ordered set. A bi-dimensional Kripke frame (or, simply, a frame) is an object of the type $\mathcal{F} = \mathbb{D} \times \mathbb{D}$. Fixed a set $\mathcal{P}$ of propositional letters, a bi-dimensional Kripke structure (structure) $\mathcal{M}$ is a bi-dimensional Kripke frame enriched with a valuation function $\mathcal{V} : D \times D \rightarrow 2^{\mathcal{P}}$, that is, $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$. The truth of a formula φ on a bi-dimensional model $\mathcal{M}$ at a point (i, j) is given by the following recursive rules:

$$\begin{aligned} \mathcal{M}, (i, j) \models p &\text{ iff } p \in \mathcal{V}(i, j); \\ \mathcal{M}, (i, j) \models \neg \varphi &\text{ iff } \mathcal{M}, (i, j) \not\models \varphi; \\ \mathcal{M}, (i, j) \models \varphi \wedge \psi &\text{ iff } \mathcal{M}, (i, j) \models \varphi \text{ and } \mathcal{M}, (i, j) \models \psi; \\ \mathcal{M}, (i, j) \models \diamondleft \varphi &\text{ iff there is } k < i \text{ s.t. } \mathcal{M}, (k, j) \models \varphi; \\ \mathcal{M}, (i, j) \models \diamondright \varphi &\text{ iff there is } k > i \text{ s.t. } \mathcal{M}, (k, j) \models \varphi; \\ \mathcal{M}, (i, j) \models \diamonddown \varphi &\text{ iff there is } k < j \text{ s.t. } \mathcal{M}, (i, k) \models \varphi; \\ \mathcal{M}, (i, j) \models \diamondup \varphi &\text{ iff there is } k > j \text{ s.t. } \mathcal{M}, (i, k) \models \varphi. \end{aligned}$$

CL has an obvious geometric interpretation, in which the operators can be interpreted as *to the north, south, east, and west*; it also has an equally obvious bi-temporal interpretation in which the operators can be interpreted as *in the future* and *past* along two orthogonal temporal dimensions. Satisfiability of formulas of CL can be studied for several classes of bi-dimensional frames. Here, we are interested in the classes of (strongly) discrete frames, such as the class of all finite frames and the class of all frames based on the set of natural/integer numbers. In general, we shall assume that D is either finite of cardinality $2 \cdot N + 1$ (from $-N$ to N), and in this case we say that $\mathcal{M}$ has *size* N (denoted $|\mathcal{M}| = N$), or $D = \mathbb{Z}$. From now on, we focus on the satisfiability of a CL -formula at the point $(0, 0)$ of some discrete model, and we say that φ is (*finitely*) *satisfiable* if and only if it is satisfied on some (finite) discrete structure $\mathcal{M}$ at the point $(0, 0)$ (we also refer to this case as the *finite/discrete* case); the *satisfiability problem* for CL is the problem of establishing if a given formula is satisfiable, and it is denoted by $SAT(CL)$.

There are several additional operators that can be expressed in the language of basic CL , and that will be useful later. First, four *partial universal* modalities, ensuring a formula to be true everywhere up-right (resp., up-left, bottom-right, bottom-left) of the current point, excluding it (*irreflexive* version):

$$\begin{aligned} \Boxup \varphi &\equiv \Boxright \varphi \wedge \Boxdown \varphi \wedge \Boxleft \varphi, & \Boxright \varphi &\equiv \Boxup \varphi \wedge \Boxdown \varphi \wedge \Boxleft \varphi, \\ \Boxdown \varphi &\equiv \Boxleft \varphi \wedge \Boxright \varphi \wedge \Boxup \varphi, & \Boxleft \varphi &\equiv \Boxdown \varphi \wedge \Boxright \varphi \wedge \Boxup \varphi, \end{aligned}$$

and including it (*reflexive* version):

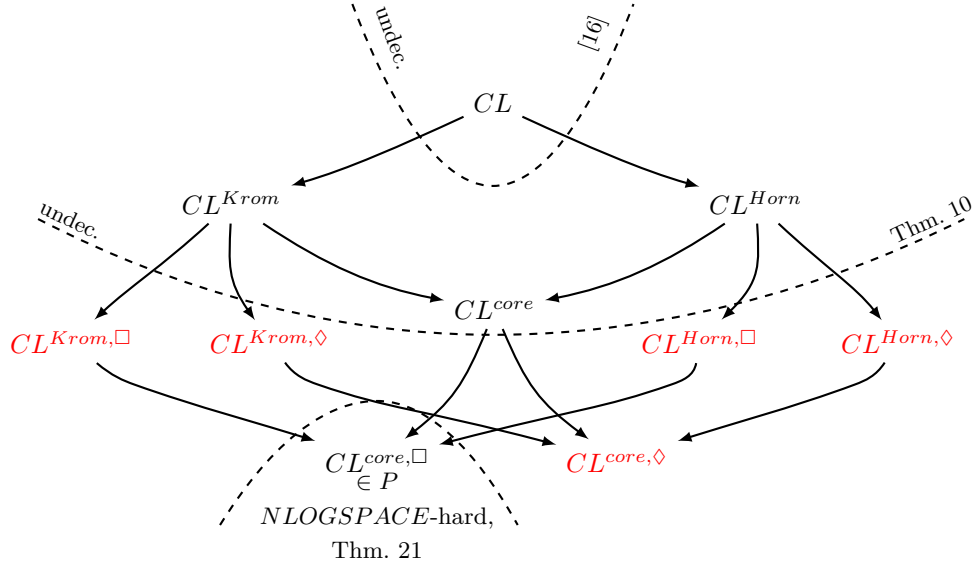
$$\begin{aligned} \Boxup \varphi &\equiv \varphi \wedge \Boxup \varphi, & \Boxright \varphi &\equiv \varphi \wedge \Boxright \varphi, \\ \Boxdown \varphi &\equiv \varphi \wedge \Boxdown \varphi, & \Boxleft \varphi &\equiv \varphi \wedge \Boxleft \varphi. \end{aligned}$$

Then, a *bidirectional* universal operator, placing φ everywhere on the same horizontal line, vertical line, and cross, excluding the current point (*irreflexive* version):

$$\Box \varphi \equiv \Boxup \varphi \wedge \Boxdown \varphi, \quad \Box \varphi \equiv \Boxright \varphi \wedge \Boxleft \varphi, \quad \Box \varphi \equiv \Boxup \varphi \wedge \Boxdown \varphi,$$

and including it (*reflexive* version):

$$\Box \varphi \equiv \varphi \wedge \Box \varphi, \quad \Box \varphi \equiv \varphi \wedge \Box \varphi, \quad \Box \varphi \equiv \varphi \wedge \Box \varphi.$$



■ **Figure 2** Relative expressive power by containment and known complexity results of fragments of CL in the finite/discrete case. The status of red fragments is unknown.

Finally, the classical (*full*) *universal* operator, which ensures that φ holds everywhere in a model:

$$\blacksquare\varphi \equiv \square\varphi \wedge \square\square\varphi \wedge \square\square\square\varphi \wedge \square\square\square\square\varphi.$$

All newly defined operators have, of course, an existential counterpart.

3 Sub-Propositional Fragments of Compass Logic

One way to define sub-propositional fragments of modal logics, that is, fragments of modal logic whose propositional expressive power is lower than that of standard Boolean logic, has been suggested in [9].

► **Definition 3.** A formula of CL is said to be in clausal form if and only if it is produced by the following grammar:

$$\begin{aligned} \lambda &::= \top \mid p \mid \Diamond\lambda \mid \square\lambda \\ C &::= \neg\lambda_1 \vee \dots \vee \neg\lambda_n \vee \lambda_{n+1} \vee \dots \vee \lambda_{n+m} \mid \square C \\ \varphi &::= C \mid \varphi \wedge \varphi, \end{aligned}$$

where $\square \in \{\square, \square\square, \square\square\square, \square\square\square\square\}$, and $n, m \in \mathbb{N}$. We use $\square^*$ in front of a clause to denote an arbitrary finite number of repetitions of some symbol $\square$.

Objects of the type λ and $\neg\lambda$ are called *literals* (this concept in fact generalizes the idea of literal in the propositional language) and objects of the type $\square^*(\dots)$ are called *clauses*. Given a CL -formula φ , let Λ_φ be the set of all literals that are sub-formulas of φ and their negations. Observe that, when in the clausal form, a formula φ is essentially a set (i.e., a conjunction) of clauses; a clause C can always be represented in its implicative form, which we shall always adopt. While there exist alternative definitions of modal clausal forms, as argued in [9] the previous one is the most general one. Some authors prefer clauses to be always universally

quantified, in the spirit of logic programming; since the universal operator is definable in the language, and since by adding enough fresh propositional letters we can always simulate the behaviour of a non-universal clause with universal ones, the two definitions are in fact equivalent up to satisfiability.

The clausal form of modal logic is Boolean-complete; however, it allows one to identify sub-Boolean, or *sub-propositional*, fragments, as follows.

► **Definition 4.** A *CL-clause*

$$\Box^*(\neg\lambda_1 \vee \dots \vee \neg\lambda_n \vee \lambda_{n+1} \vee \dots \vee \lambda_{n+m})$$

is said to be *Horn* if and only if $m \leq 1$, *Krom* if and only if $n + m \leq 2$, and *core* if it is both *Horn* and *Krom*.

The Horn, Krom, and core fragments of CL are denoted, respectively, CL^{Horn} , CL^{Krom} , and CL^{core} . While in the full Boolean language universal and existential operators are mutually definable, this, in general, is no longer true in sub-propositional ones. Thus, for $\spadesuit \in \{Horn, Krom, core\}$ it makes sense to define and study the fragments $CL^{\spadesuit, \Box}$, allowing only the universal version of the operators, and $CL^{\spadesuit, \Diamond}$, allowing only the existential one. These restrictions give rise to the natural taxonomy in Fig. 2. For a fragment L of CL , the corresponding satisfiability problem is denoted by $SAT(L)$.

Sub-propositional fragments may come across as too weak to be useful, especially when devoid of existential operators. However, when CL is seen as a bi-temporal logic, one could take advantage of its ability of expressing statements on two orthogonal dimensions, such as valid and transaction time in a temporal database, and formalize interesting statements. Just to limit ourselves to a single example, consider a temporal database with employees' information, and the case of a fired employee, for which the information *fired* should be persistent, that is, once someone is fired he/she is never hired back, and the system should never forget about it:

$$\blacksquare(fired \rightarrow (\Box fired \wedge \Box fired)).$$

In fact, one could express also interesting situations such as those modeled in [8, 10], at least in some reduced form. More in general, the fragment $L^{Horn, \Box}$ of a propositional modal logic L , and therefore its fragment $L^{core, \Box}$, translates to the Horn fragment of first-order logic via standard translation, underlying its naturalness, and, as it turns out, on top of the classical fields of automatic theorem proving and logic programming Horn fragments are often considered in recent contributions that focus on machine learning [4, 13], and having well-behaved fragments of bi-dimensional logic under Horn-like restrictions allows one to imagine similar applications specialized to the bi-dimensional case.

In the rest of this paper, focusing on finite/discrete structures only, we study the decidability of the satisfiability problem of sub-propositional fragments of CL, and we show that for CL^{core} this problem is still undecidable, while it becomes decidable for $CL^{core, \Box}$.

4 Undecidability of CL^{core}

The idea of the proof stems from the original undecidability argument for the non-clausal case [16]; we shall use a similar schema, and partially inherit the notation. Unlike the original result, however, we prove that the satisfiability of CL^{core} is undecidable via reduction from the acceptance problem of a deterministic Turing machine with empty input, rather than

$$ACCEPTS \leq_m SAT(CL^{core}).$$

About shortcuts in core clauses. Three considerations should be made. First, core (resp., Horn) clauses with a leading $\Box^*$, where $\Box \in \{\Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box, \Box\}$ can always be translated into equivalent conjunctions of core (resp., Horn) each with a leading $\Box^*$, where $\Box \in \{\Box, \Box, \Box, \Box\}$. Second, a similar equivalence holds for their existential counterparts, when used to form literals. Third, while strictly speaking $\Box \neg \lambda$ is not a literal, it is semantically equivalent to $\neg \Diamond \lambda$, which is allowed.

Consider the following clauses:

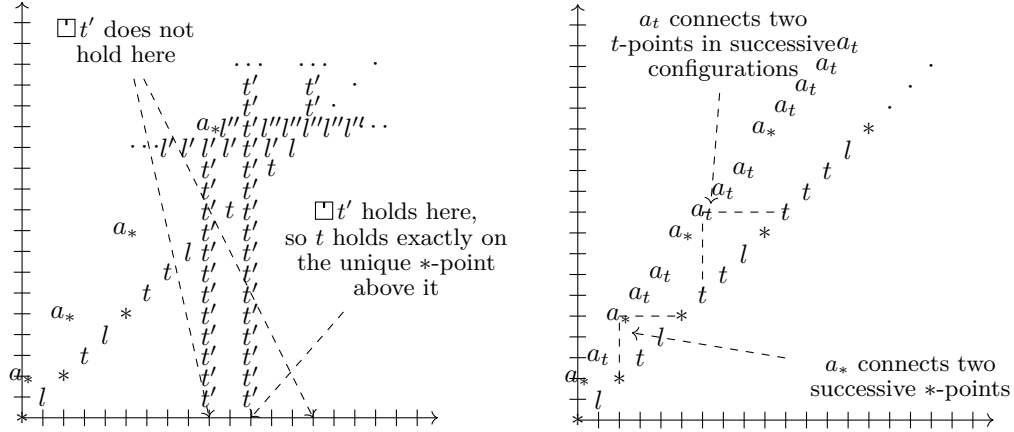
$$\Diamond e \wedge \Box(e \rightarrow \Diamond \bar{e}) \wedge p \wedge p_e \wedge \Box \neg p \wedge \blacksquare(e \rightarrow \Box \Box \Box \neg p) \wedge \blacksquare(p \rightarrow \Box \neg p), \quad (1)$$

$$\blacksquare(p_e \rightarrow \boxed{\square}\boxed{\square}\neg u) \wedge \blacksquare(p_e \rightarrow \blacklozenge u) \wedge \blacksquare(u \rightarrow \blacklozenge p), \quad (2)$$

$$\blacksquare(\diamondleft e \rightarrow \diamonddown p_e) \wedge \blacksquare(p_e \rightarrow p). \quad (3)$$

- (i) $\mathcal{M}, (i, i) \models p$ for every $0 \leq i \leq i_k$;
- (ii) $\mathcal{M}, (i, i+1) \models u$ for every $0 \leq i \leq i_k$, and
- (iii) $\mathcal{M}, (i, j) \not\models p$ for any other point (i, j) .

$$\mathcal{L} = \{*, 0, 1, \sqcup, f, t, l, g\} \cup \bigcup_{q \in Q, c \in \{0, 1, \sqcup\}, v \in \{f, g\}} \{q, q_c, q_{c,v}, r_c\};$$
$$\bigwedge_{z \in \mathcal{L}} \blacksquare(z \rightarrow p). \quad (4)$$



■ **Figure 3** To the **left**: a partial representation of a model satisfying $(1) \wedge \dots \wedge (15)$. Observe, in particular, how l'' holds to the right of a a_* -point, and l' to the left of a l -point; since the former implies that t' holds everywhere above it (and including it), and the latter implies that t' holds everywhere below it (and including it), every point at height 0 whose horizontal coordinate is strictly between that of an a_* and that of an l is such that t' holds everywhere above it, and this, in turn, implies that t holds somewhere above it. To the **right**: a partial representation of a model satisfying $(1) \wedge \dots \wedge (24)$. By way of example, observe how a $*$ -point is connected to the next one and how a t -point is connected to the next one.

Finally, taking advantage of the sequence of p -points so defined, we can also use:

$$\Diamond z \equiv \Diamond z' \wedge \Box(z' \rightarrow u) \wedge \Box(z' \rightarrow \Diamond z) \wedge \Box(z \rightarrow p)$$

and

$$\Diamond z \equiv \Diamond z' \wedge \Box(z' \rightarrow u) \wedge \Box(z' \rightarrow \Diamond z) \wedge \Box(z \rightarrow p)$$

on p -points to place a propositional letter z on the next (resp., previous) p -point.

Preparing the sequence of configurations. Now, we define the notion of machine configuration as a sequence of p -points between two consecutive $*$ -points. We use the letter a_* to connect one $*$ -point to the next one, using e and the new letters $*_e$ and e_* to avoid forcing an infinite sequence. Thus, we have:

$$* \wedge \blacksquare(\Diamond e \rightarrow e), \quad (5)$$

$$\Box(e \rightarrow \Box e_*) \wedge \Box \Box(e \rightarrow \Box e_*) \wedge \blacksquare(* \rightarrow \Box e_*), \quad (6)$$

$$*_e \wedge \Box(\Box e_* \rightarrow \Diamond *_e) \wedge \blacksquare(*_e \rightarrow p), \quad (7)$$

$$\blacksquare(*_e \rightarrow \Diamond a_*) \wedge \blacksquare(a_* \rightarrow \Diamond *), \quad (8)$$

$$\Box(* \rightarrow \Diamond a_*) \wedge \blacksquare(a_* \rightarrow \Diamond *), \quad (9)$$

$$\blacksquare(a_* \rightarrow \Box \Box \neg *) \wedge \blacksquare(a_* \rightarrow \Box \Box \neg a_*) \wedge \blacksquare(a_* \rightarrow \Box \neg a_*), \quad (10)$$

$$\Diamond l \wedge \Box(* \rightarrow \Diamond l) \wedge \blacksquare(l \rightarrow \Diamond *), \quad (11)$$

$$\blacksquare(l \rightarrow \Box l') \wedge \blacksquare(a_* \rightarrow \Box l''), \quad (12)$$

$$\blacksquare(l' \rightarrow \Box t') \wedge \blacksquare(l'' \rightarrow \Box t'), \quad (13)$$

$$\blacksquare(l' \rightarrow t') \wedge \blacksquare(l'' \rightarrow t'), \quad (14)$$

$$\Box(\Box t' \rightarrow \Diamond t). \quad (15)$$

Fig. 3 (left-hand side) shows a partial representation of this part of the construction.

The first configuration is always constituted by a single tape cell, and the s -th configuration must be exactly one tape cell longer than the $(s - 1)$ -th. To achieve this, similarly to what already done with $*$ -points, t - and l -points belonging to a configuration too will be connected to their corresponding ones in the next configuration, and this connection is mediated by the letter a_t . As before, such connections cannot force a model to be infinite; thus, t - (resp., l -) points that are below the level of e are marked by t_e (resp., l_e), and such a construction makes use of the auxiliary letter e_t (resp., e_l) to carry the information of the position of a t (resp., l); from an abstract point of view, the construction that makes this possible is quite similar to that used before to connect successive $*$ -points. This is realized by saying:

$$\blacksquare(e \rightarrow \Box e_t) \wedge \blacksquare(t \rightarrow \Box e_t), \quad (16)$$

$$\blacksquare(e \rightarrow \Box e_l) \wedge \blacksquare(l \rightarrow \Box e_l), \quad (17)$$

$$\Box(\Box e_t \rightarrow \Diamond t_e) \wedge \blacksquare(t_e \rightarrow t), \quad (18)$$

$$\Box(\Box e_l \rightarrow \Diamond l_e) \wedge \blacksquare(l_e \rightarrow l), \quad (19)$$

$$\blacksquare(t_e \rightarrow \Diamond a_t) \wedge \blacksquare(l_e \rightarrow \Diamond a_t) \wedge \blacksquare(a_t \rightarrow \Diamond t), \quad (20)$$

$$\blacksquare(t \rightarrow \Diamond a_t), \quad (21)$$

$$\blacksquare(a_t \rightarrow \Box \neg a_t) \wedge \blacksquare(a_t \rightarrow \Box \neg a_t), \quad (22)$$

$$\blacksquare(a_* \rightarrow \Box \neg a_t) \wedge \blacksquare(a_t \rightarrow \Box \neg a_*) \wedge \blacksquare(a_t \rightarrow \Diamond \Diamond *), \quad (23)$$

$$\bigwedge_{z_1, z_2 \in \{l, t, *\}, z_1 \neq z_2} \blacksquare(z_1 \rightarrow \neg z_2). \quad (24)$$

Fig. 3 (right-hand side) shows a partial representation of this part of the construction.

► **Lemma 6.** *Suppose that $\mathcal{M}, (0, 0) \models (1) \wedge \dots \wedge (24)$, and let $(i_0, i_0), \dots, (i_k, i_k)$ be the sequence of p -points whose existence is guaranteed by Lem. 5. Then:*

- (i) *there exists a finite sequence $(j_0, j_0), (j_1, j_1), \dots$ of $*$ -points on top of the sequence $(i_0, i_0), \dots, (i_k, i_k)$ whose existence is guaranteed by Lem. 5, no other points are $*$ -points, and $j_s \leq k$ for every s ;*
- (ii) *for each $s \geq 2$, $(j_s - 1, j_s - 1)$ is an l -point, and no other points are l -points;*
- (iii) *for each $s \geq 1$, all points between $(j_s + 1, j_s + 1)$ and $(j_{s+1} - 2, j_{s+1} - 2)$, included, are t -points, and no other points are t -points, and*
- (iv) *for each $s \geq 2$, $j_{s+1} - j_s = j_s - j_{s-1} + 1$.*

We say that every $$ -point starts a configuration.*

Introducing the transition function. Now we encode the transition function. To this end, we re-label tape cells in order to separate the first tape cell of each configuration (denoted with f) from any other tape cell (denoted by g). This construction is no different from the one already used to label all p -points between a $*$ -point and its corresponding l -point, and makes use of the auxiliary symbols f' , f'' , g' , and b_* :

$$\blacksquare(* \rightarrow \Diamond f), \quad (25)$$

$$\blacksquare(* \rightarrow \Diamond b_*) \wedge \blacksquare(b_* \rightarrow \Diamond *) \wedge \blacksquare(b_* \rightarrow \Box \neg *), \quad (26)$$

$$\blacksquare(f \rightarrow \Box f') \wedge \blacksquare(b_* \rightarrow \Box f''), \quad (27)$$

$$\blacksquare(f' \rightarrow \Box g') \wedge \blacksquare(f'' \rightarrow \Box g'), \quad (28)$$

$$\Box(\Box g' \rightarrow \Diamond g) \wedge \blacksquare(f \rightarrow \neg g) \wedge \blacksquare(g \rightarrow \neg f). \quad (29)$$

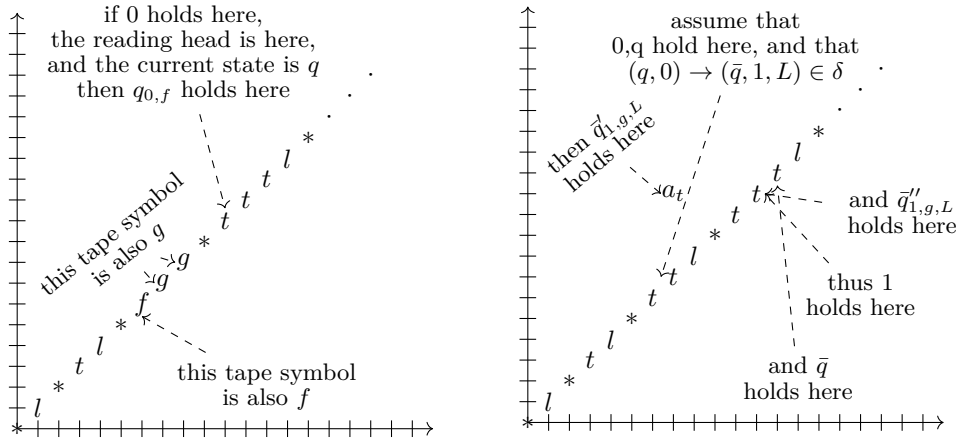


Figure 4 To the **left**: a partial representation of a model that satisfying $(1) \wedge \dots \wedge (37)$. By way of example, assume that the fourth configuration starts with 0, in the state q , and with the reading head on the first tape cell; then, we encode this situation by making sure that $q_{0,f}$ holds on that tape cell too. To the **right**: a partial representation of a model that satisfies the formula $(1) \wedge \dots \wedge (47)$: observe how one combined letter transfers the information to the next t -point using the auxiliary letter a_t .

► **Lemma 7.** *Suppose that $\mathcal{M}, (0, 0) \Vdash (1) \wedge \dots \wedge (29)$, and let $(j_0, j_0), (j_1, j_1), \dots$ be the sequence of configurations whose existence is guaranteed by Lem. 6. Then:*

- (i) for each $s \geq 1$, $(j_s + 1, j_s + 1)$ is a f -point, and no other points are f -points;
- (ii) for each $s \geq 1$, all points between $(j_s + 2, j_s + 2)$ and $(j_{s+1} - 1, j_{s+1} - 1)$, included, are g -points, and no other points are g -points.

In the reduction the actual tape symbols, denoted here by the letters 0, 1, and $\sqcup$, label f -, l -, and t -points, and they must be placed according to the transition function from one configuration to the next one. For each state $q \in Q$ we use a letter q to denote both the state in which the machine is in a particular configuration and the letter denoting it; the particular t -point on which q holds also denotes the tape symbol being read. Moreover, we use $q_0, q_1, q_{\sqcup}$ (resp., $r_0, r_1, r_{\sqcup}$) to denote the combined information of the machine reading 0, 1, or $\sqcup$ and being in the state q (resp., the machine not reading that particular tape position which is holding 0, 1, or $\sqcup$). Finally, we use the subscript f or g to take into account the fact that the position being read is the first one of the configuration or it is not. In the rest of this construction, q ranges in Q , c ranges in $\{0, 1, \sqcup\}$, and v ranges in $\{f, g\}$, to form the sets of letters $q_{c,v}$ and r_c ; for example, $q_{0,f}$ holds on some p -point only when both q and 0 hold on it and the position of the cell is the first one in the configuration.

Stepping from individual tape symbols, state symbols, and positions, to combined symbols (for example, from a t -point that also satisfies 0, q , and it is the first one in a configuration to having $q_{0,f}$ holding on the same t -point) is achieved as follows. In correspondence of every tape symbol z , an auxiliary letter z' is placed somewhere below it. Then, depending on the particular symbol currently placed on a particular t -point, such an auxiliary letter is forced to appear above the horizontal line (when it is 0), below it (when it is 1), or on it (when it is $\sqcup$), and this information is later used to decide what combined letter is placed on that particular t -point. For combined letters of the type r_c , this construction is enough, as we do not need to codify their relative position within the configuration. To combined letter of the type q_c , on the other hand, we need to add the position information; we do so via

the auxiliary letters q'_c , that are forced to appear somewhere below each occurrence of a q_c , and particularly strictly above the horizontal line for first positions and strictly below it for non-first positions, and depending on where they actually occur, the final combined letters of the type $q_{c,f}$ and $q_{c,g}$ are built and placed on the same t -point. This construction is exemplified in Fig. 4 (left-hand side). We proceed by asserting:

$$\bigwedge_{z \in Q \cup \{r\}} (\Box(\Diamond 0 \rightarrow \neg z') \wedge \Box(\Diamond 0 \rightarrow \Box \neg z')), \quad (30)$$

$$\bigwedge_{z \in Q \cup \{r\}} (\Box(\Diamond 1 \rightarrow \neg z') \wedge \Box(\Diamond 1 \rightarrow \Box \neg z')), \quad (31)$$

$$\bigwedge_{z \in Q \cup \{r\}} (\Box(\Diamond \sqcup \rightarrow \Box \neg z') \wedge \Box(\Diamond \sqcup \rightarrow \Box \neg z')), \quad (32)$$

$$\bigwedge_{z \in Q \cup \{r\}} \blacksquare(z \rightarrow \Diamond z'), \quad (33)$$

$$\bigwedge_{z \in Q \cup \{r\}} (\Box(z' \rightarrow \Diamond z_\sqcup) \wedge \Box(\Diamond z' \rightarrow \Diamond z_0) \wedge \Box(\Diamond z' \rightarrow \Diamond z_1)), \quad (34)$$

$$\bigwedge_{q \in Q, c \in \{0,1,\sqcup\}} (\Box \neg q'_c \wedge \Box(\Diamond f \rightarrow \Box \neg q'_c) \wedge \Box(\Diamond g \rightarrow \Box \neg q'_c)), \quad (35)$$

$$\bigwedge_{q \in Q, c \in \{0,1,\sqcup\}} (\blacksquare(q_c \rightarrow \Diamond q'_c)), \quad (36)$$

$$\bigwedge_{q \in Q, c \in \{0,1,\sqcup\}} (\Box(\Diamond q'_c \rightarrow q_{c,f}) \wedge \Box(\Diamond q'_c \rightarrow q_{c,g})). \quad (37)$$

► **Lemma 8.** *If $\mathcal{M}, (0,0) \models (1) \wedge \dots \wedge (37)$ and $(i_0, i_0), (i_1, i_1), \dots, (i_k, i_k)$ is the sequence of p -points whose existence is guaranteed by Lem. 5, then:*

- (i) *if (i_s, i_s) satisfies q for some $q \in Q$, c for some $c \in \{0,1,\sqcup\}$, and f (resp., g), then it also satisfies $q_{c,f}$ (resp., $q_{c,g}$);*
- (ii) *if (i_s, i_s) satisfies r and c for some $c \in \{0,1,\sqcup\}$, then it also satisfies r_c .*

The last part of the reduction requires forcing configurations to behave according to the transition function δ . This implies making sure that the first configuration represents the Turing machine $\mathcal{A}$ starting with a 1-symbol tape holding $\sqcup$, the reading head currently on the first position, and the machine itself in state q_i . Reasoning by induction, and assuming that a particular configuration holds a particular sequence of tape symbols (on t -points) and that a particular letter q (for $q \in Q$) holds on some t -point (indicating that the reading head is currently on that position), the previous lemma ensures that all t -points consistently satisfy the letters $q_{c,f}$, $q_{c,g}$, and r_c , for $c \in \{0,1,\sqcup\}$. Our task is now that of transferring the information to the next configuration, in terms of symbols $0, 1, \sqcup$ and q . This is achieved by means of the auxiliary letter a_t (introduced in Lem. 6) and the auxiliary letters of the type $q'_{c,f,M}$, with $M \in \{R, L\}$ (to denote a right or left movement of the reading head) and r'_c , that are forced to hold on a a_t point above some t -point and transfer the information on the corresponding t -point on its right, as well as $q''_{c,f,M}$, r''_c , that hold on t -points. Separating the situations in which the reading head is sitting, or not, on the first position in a configuration allows us to respond correctly to the particular case in which δ requires a left movement, which, in that case, must be ignored. An example is shown in Fig. 4 (right-hand side). Thus, we set:

$$\Diamond q_i \wedge \Diamond \sqcup \wedge \bigwedge_{q \in Q} \blacksquare(q \rightarrow \neg *), \quad (38)$$

$$\bigwedge_{(q,c) \rightarrow (\bar{q}, \bar{c}, M) \in \delta, v \in \{f, g\}} \blacksquare(q_{c,v} \rightarrow \Diamond \bar{q}'_{\bar{c},v,M}) \wedge \bigwedge_{c \in \{0,1,\sqcup\}} \blacksquare(r_c \rightarrow \Diamond \bar{r}'_c), \quad (39)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, M \in \{R, L\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}'_{\bar{c},v,M} \rightarrow \Diamond \bar{q}''_{\bar{c},v,M}) \wedge \blacksquare(\bar{r}'_c \rightarrow \Diamond \bar{r}''_c)), \quad (40)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, M \in \{R, L\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}'_{\bar{c},v,M} \rightarrow a_t) \wedge \blacksquare(\bar{r}'_{\bar{c}} \rightarrow a_t)), \quad (41)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, M \in \{R, L\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}''_{\bar{c},v,M} \rightarrow t) \wedge \blacksquare(\bar{r}''_{\bar{c}} \rightarrow t)), \quad (42)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}''_{\bar{c},v,R} \rightarrow \Diamond \bar{q}) \wedge \blacksquare(\bar{q}''_{\bar{c},v,R} \rightarrow \bar{c})), \quad (43)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}''_{\bar{c},g,L} \rightarrow \Diamond \bar{q}) \wedge \blacksquare(\bar{q}''_{\bar{c},g,L} \rightarrow \bar{c})), \quad (44)$$

$$\bigwedge_{q \in Q, v \in \{f, g\}, c \in \{0,1,\sqcup\}} (\blacksquare(\bar{q}''_{\bar{c},f,L} \rightarrow \bar{q}) \wedge \blacksquare(\bar{q}''_{\bar{c},f,L} \rightarrow \bar{c})), \quad (45)$$

$$\bigwedge_{c \in \{0,1,\sqcup\}} \blacksquare(\bar{r}''_c \rightarrow c), \quad (46)$$

$$\bigwedge_{c_1, c_2 \in \{0,1,\sqcup\}} \blacksquare(c_1 \rightarrow \neg c_2) \wedge \bigwedge_{q_1, q_2 \in Q} \blacksquare(q_1 \rightarrow \neg q_2). \quad (47)$$

► **Lemma 9.** *If $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_i, q_a, q_r)$ is a deterministic Turing machine, $\mathcal{M}, (0, 0) \models (1) \wedge \dots \wedge (47)$, and $(j_0, j_0), (j_1, j_1), \dots$ is the sequence of $*$ -points whose existence is guaranteed by Lem. 6, then, for every $n \geq 0$ and $m \geq 1$:*

- (i) *if $(j_s + d, j_s + d)$ satisfies q for some $q \in Q$, c for some $c \in \{0, 1, \sqcup\}$, and f (resp., g), and if $(q, c) \rightarrow (\bar{q}, \bar{c}, R) \in \delta$, then $(j_{s+1} + d, j_{s+1} + d)$ satisfies $\bar{c}$ and $(j_{s+1} + d + 1, j_{s+1} + d + 1)$ satisfies $\bar{q}$;*
- (ii) *if $(j_s + d, j_s + d)$ satisfies q for some $q \in Q$, c for some $c \in \{0, 1, \sqcup\}$, and g , and if $(q, c) \rightarrow (\bar{q}, \bar{c}, L) \in \delta$, then $(j_{s+1} + d, j_{s+1} + d)$ satisfies $\bar{c}$ and $(j_{s+1} + d - 1, j_{s+1} + d - 1)$ satisfies $\bar{q}$;*
- (iii) *if $(j_s + d, j_s + d)$ satisfies q for some $q \in Q$, c for some $c \in \{0, 1, \sqcup\}$, and f , and if $(q, c) \rightarrow (\bar{q}, \bar{c}, L) \in \delta$, then $(j_{s+1} + d, j_{s+1} + d)$ satisfies both $\bar{c}$ and $\bar{q}$;*
- (iv) *if $(j_s + d, j_s + d)$ does not satisfy q for any $q \in Q$, and satisfies c for some $c \in \{0, 1, \sqcup\}$, then $(j_{s+1} + d, j_{s+1} + d)$ satisfies c .*

Completing the reduction. Consider now the formula:

$$(1) \wedge \dots \wedge (47) \wedge \Diamond \Diamond q_a. \quad (48)$$

► **Theorem 10.** *Given a Turing machine $\mathcal{A}$, the CL^{core} -formula (48) is satisfiable on a finite/discrete structure if and only if $\mathcal{A}$ accepts on the empty input. Therefore, the satisfiability problem for CL^{core} in the finite/discrete case is undecidable.*

5 Decidability of $CL^{core, \square}$

In this section, we prove that excluding diamonds from the grammar of the core fragment of CL entails regaining the decidability of its satisfiability problem in classes of discrete frames, as an effect of the low expressive power of this language.

Minimal extended structures. Let us start by introducing the concept of minimal extended structure.

► **Definition 11.** *Given a bi-dimensional Kripke frame $\mathcal{F}$, a bi-dimensional extended Kripke structure (or, simply, extended structure) $\mathcal{M}^+$ is a bi-dimensional Kripke frame enriched with an extended valuation function $\mathcal{V}^+ : D \times D \rightarrow 2^{\Lambda_\varphi}$, that is, $\mathcal{M}^+ = \langle \mathcal{F}, \mathcal{V}^+ \rangle$, such that, for every point $(i, j) \in \mathcal{F}$:*

- $\neg\lambda \in \mathcal{V}^+(i, j)$ iff $\lambda \notin \mathcal{V}^+(i, j)$,
- $\Box\lambda \in \mathcal{V}^+(i, j)$ iff $\lambda \in \mathcal{V}^+(k, j)$ for every $k < i$,
- $\Box\lambda \in \mathcal{V}^+(i, j)$ iff $\lambda \in \mathcal{V}^+(k, j)$ for every $k > i$,
- $\Box\lambda \in \mathcal{V}^+(i, j)$ iff $\lambda \in \mathcal{V}^+(i, k)$ for every $k > j$,
- $\Box\lambda \in \mathcal{V}^+(i, j)$ iff $\lambda \in \mathcal{V}^+(i, k)$ for every $k < j$.

The notion of a structure $\mathcal{M}$ satisfying a formula φ can be lifted to a notion of an extended structure $\mathcal{M}^+$ satisfying the formula φ by simply replacing the base case in Def. 2 with:

$$\mathcal{M}^+, (i, j) \models \lambda \text{ iff } \lambda \in \mathcal{V}^+(i, j).$$

The following result is an immediate consequence of the definition.

► **Lemma 12.** *Let φ be a $CL^{core, \Box}$ -formula. Then, φ is satisfiable if and only if it is satisfied on some extended structure.*

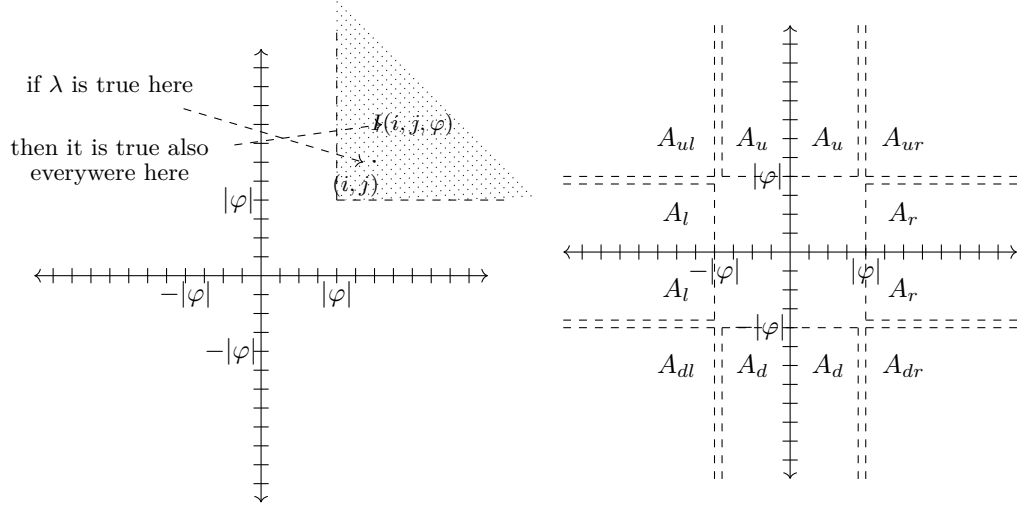
Since satisfiability of $CL^{core, \Box}$ -formulas can be reduced to satisfiability on extended structures, we use the same notation and terminology for structures (resp., models) and extended structures (resp., models).

► **Definition 13.** *Given a structure $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$, a point (i, j) , and a literal λ such that $\lambda \in \mathcal{V}(i, j)$, we say that a (i, j, λ) -reduct of $\mathcal{M}$ is any structure $\mathcal{M}' = \langle \mathcal{F}, \mathcal{V}' \rangle$ such that $\lambda \notin \mathcal{V}'(i, j)$. A structure $\mathcal{M}$ is said to be minimal for a $CL^{core, \Box}$ -formula φ if and only if $\mathcal{M}, (0, 0) \models \varphi$ and for every (i, j, λ) -reduct $\mathcal{M}'$ of $\mathcal{M}$ it is the case that $\mathcal{M}', (0, 0) \not\models \varphi$.*

► **Lemma 14.** *Let φ be a $CL^{core, \Box}$ -formula. Then, φ is satisfiable if and only if it is satisfied by some minimal structure.*

Minimal structures are a standard notion in Horn-like fragments of propositional (modal) logic, based on the observation that in such fragments there cannot be a disjunction of positive literals; this allows one to partially order structures by containment, which, in spatial logic, becomes pointwise containment. Since satisfiability of $CL^{core, \Box}$ -formulas can be reduced to satisfiability on minimal structures, again, we make the notation and terminology uniform by using structure (resp., model) instead of minimal extended structure (resp., minimal extended model).

Ultimately constant structures. In order to understand the properties of (minimal extended) structures that satisfy $CL^{core, \Box}$ -formulas, first we prove that positive literals cannot exist in isolated points except on $(0, 0)$.



■ **Figure 5** To the **left**: a pictorial representation of influence area and Lem. 16. To the **right**: a pictorial representation of an ultimately constant model and Lem. 19.

► **Definition 15.** Let φ be a $CL^{core, \square}$ -formula, $\mathcal{F}$ a structure, $(i, j) \in \mathcal{F}$ a point. The influence area of (i, j) w.r.t. φ , denoted $I(i, j, \varphi)$, is defined by the following rule. For every (i', j') , it is the case that $(i', j') \in I(i, j, \varphi)$ if and only if:

$$\begin{array}{ll}
 i' \geq \min\{i, |\varphi|\}, j' \geq \min\{j, |\varphi|\}, & \text{if } i > 0, j > 0, \\
 i' = 0, j' \geq \min\{j, |\varphi|\}, & \text{if } i = 0, j > 0, \\
 i' \leq \max\{i, -|\varphi|\}, j' \geq \min\{j, |\varphi|\}, & \text{if } i < 0, j > 0, \\
 i' \leq \max\{i, -|\varphi|\}, j' = 0, & \text{if } i < 0, j = 0, \\
 i' \leq \max\{i, -|\varphi|\}, j' \leq \max\{j, -|\varphi|\}, & \text{if } i < 0, j < 0, \\
 i' = 0, j' \leq \max\{j, -|\varphi|\}, & \text{if } i = 0, j < 0, \\
 i' \geq \min\{i, |\varphi|\}, j' \leq \max\{j, -|\varphi|\}, & \text{if } i > 0, j < 0, \\
 i' \geq \min\{i, |\varphi|\}, j' = 0, & \text{if } i > 0, j = 0.
 \end{array}$$

► **Lemma 16.** Let φ be a $CL^{core, \square}$ -formula, $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$ a model for φ , and λ a literal. Then, if $\mathcal{M}, (i, j) \models \lambda$ for some point $(i, j) \neq (0, 0)$, then $\mathcal{M}, (i', j') \models \lambda$ for every $(i', j') \in I(i, j, \varphi)$.

A pictorial representation of the effects of Lem. 16 can be found in Fig. 5 (left-hand side).

Now, when we look at points far away from the point $(0, 0)$ we see that structures have a very well-defined shape.

► **Definition 17.** Let φ be a $CL^{core, \square}$ -formula, and $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$ a model for φ . The up (resp., up right, right, down right, down, down left, left, up left) area, denoted $A_{\clubsuit}$, for $\clubsuit \in \{u, ur, r, dr, d, dl, l, ul\}$, is defined as follows:

$$\begin{array}{l}
 A_u = \{(i, j) \mid -|\varphi| < i < |\varphi|, j \geq |\varphi|\}; \\
 A_{ur} = \{(i, j) \mid i \geq |\varphi|, j \geq |\varphi|\}; \\
 A_r = \{(i, j) \mid i \geq |\varphi|, -|\varphi| < j < |\varphi|\}; \\
 A_{dr} = \{(i, j) \mid i \geq |\varphi|, j \leq -|\varphi|\}; \\
 A_d = \{(i, j) \mid -|\varphi| < i < |\varphi|, j \leq -|\varphi|\}; \\
 A_{dl} = \{(i, j) \mid i \leq -|\varphi|, j \leq -|\varphi|\}; \\
 A_l = \{(i, j) \mid i \leq -|\varphi|, -|\varphi| < j < |\varphi|\}; \\
 A_{ul} = \{(i, j) \mid i \leq -|\varphi|, j \geq |\varphi|\}.
 \end{array}$$

► **Definition 18.** Let φ be a $CL^{core, \square}$ -formula and $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$ a model for φ with size $|\mathcal{M}| > |\varphi|$. We say that $\mathcal{M}$ is ultimately constant if and only if for every pair of points $(i, j), (i', j')$ and every $\clubsuit \in \{ur, dr, dl, ul\}$:

1. if $(i, j), (i', j') \in A_{\clubsuit}$, then $\mathcal{V}(i, j) = \mathcal{V}(i', j')$;
2. if $(i, j) \in A_u$, then $\mathcal{V}(i, j) = \mathcal{V}(i, j + 1)$;
3. if $(i, j) \in A_r$, then $\mathcal{V}(i, j) = \mathcal{V}(i + 1, j)$;
4. if $(i, j) \in A_d$, then $\mathcal{V}(i, j) = \mathcal{V}(i, j - 1)$, and
5. if $(i, j) \in A_l$, then $\mathcal{V}(i, j) = \mathcal{V}(i - 1, j)$.

► **Lemma 19.** Let φ be a $CL^{core, \square}$ -formula. Then, if φ is satisfied on a structure $\mathcal{M}$ with size $|\mathcal{M}| > |\varphi|$, then $\mathcal{M}$ is ultimately constant.

A pictorial representation of the effects of Lem. 19 can be found in Fig. 5 (right-hand side).

Satisfiability of $CL^{core, \square}$ is in P . Decidability of satisfiability in $CL^{core, \square}$ in the discrete case is an immediate consequence of Lem. 19 and the fact that ultimately constant models are finitely representable. Decidability in the finite case is proven by observing that a given a $CL^{core, \square}$ -formula φ is finitely satisfiable if and only if it is satisfied on some minimal model of size $N = |\varphi|$.

► **Lemma 20.** Let φ be a $CL^{core, \square}$ -formula. Then, if φ is finitely satisfiable, then there exists a structure $\mathcal{M} = \langle \mathcal{F}, \mathcal{V} \rangle$, where $\mathcal{F} = \mathbb{D} \times \mathbb{D}$ and $\mathbb{D} = \langle D, < \rangle$, such that, first, $\mathcal{M}, (0, 0) \models \varphi$ and, second, $|\mathcal{M}| \leq |\varphi|$.

► **Theorem 21.** The satisfiability problem for $CL^{core, \square}$ in the class of all finite structures, the class of all infinite structures, and the class of all discrete structures, is in P and $NLOGSPACE$ -hard.

6 Conclusions

Despite the naturalness of Compass Logic (CL), its properties have been scarcely investigated, probably because it was introduced at the same time that its undecidability was proven, discouraging its use. Curiously, the same happened with Halpern and Shohams' Modal Logic of Allen's Relations (HS), which was immediately shown to be undecidable; yet, in the latter case a number of syntactical, semantical, and also clausal fragments were studied, contributing to the identification of a very complex landscape. In the case of CL, there have been only two previous attempts at finding well-behaved sub-languages.

In this paper we initiated the analysis of clausal fragments of CL, inspired by similar previous investigations. We unveiled a small, still undecidable fragment of CL, which turns out to be technically interesting due to the complexity of the undecidability proof, and an even smaller fragment of it, which, on the contrary, is decidable in deterministic polynomial time. Despite CL and HS being both bi-dimensional, their behaviour under Horn-like restrictions is different; in the case of CL, in particular, the core fragment turns out to be decidable in the discrete irreflexive case, which is precisely the most outstanding case left open in the case of the core fragment of HS. Whether the decidability of the satisfiability problem of core fragment of CL can be pushed further, say, to the Horn fragment, or whether its complexity can be pinpointed in a tighter way, are open problems at the moment.

Modal logics for qualitative reasoning have a wide range of real-life applications in modern artificial intelligence and machine learning. Decidable and efficient languages for reasoning about non-tabular data, such as temporal and spatial data, can be applied in machine learning-based applications, and good computational properties enable useful reasoning tasks, such as minimization of learned rules and checking expert-knowledge compliance.

References

- 1 Luca Aceto, Dario Della Monica, Valentin Goranko, Anna Ingólfssdóttir, Angelo Montanari, and Guido Sciavicco. A complete classification of the expressiveness of interval logics of Allen's relations: the general and the dense cases. *Acta Informatica*, 53(3):207–246, 2016. doi:10.1007/S00236-015-0231-4.
- 2 James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.
- 3 Linh A.Nguyen. On the complexity of fragments of modal logics. In *Proc. of the 5th Conference on Advances in Modal Logic*, pages 249–268. King's College Publications, 2004. URL: <http://www.aiml.net/volumes/volume5/Nguyen.ps>.
- 4 Marta Arias, José L. Balcázar, and Cristina Tirnăuță. Learning definite Horn formulas from closure queries. *Theoretical Computer Science*, 658:346–356, 2017. doi:10.1016/J.TCS.2015.12.019.
- 5 Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyashev. The complexity of clausal fragments of LTL. In *Proc. of the 19th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, volume 8312 of *Lecture Notes in Computer Science*, pages 35–52. Springer, 2013. doi:10.1007/978-3-642-45221-5_3.
- 6 Philippe Balbiani, Jean-François Condotta, and Luis Fariñas del Cerro. A model for reasoning about bidimensional temporal relations. In *Proc. of the 6th International Conference on Principles of Knowledge Representation and Reasoning*, pages 124–130, Trento, Italy, 1998. Morgan Kaufmann.
- 7 Davide Bresolin, Dario Della Monica, Angelo Montanari, Pietro Sala, and Guido Sciavicco. Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theoretical Computer Science*, 560:269–291, 2014. doi:10.1016/J.TCS.2014.03.033.
- 8 Davide Bresolin, Agi Kurucz, Emilio Muñoz-Velasco, Vladislav Ryzhikov, Guido Sciavicco, and Michael Zakharyashev. Horn fragments of the Halpern-Shoham interval temporal logic. *ACM Transactions on Computational Logic*, 18(3):22:1–22:39, 2017. doi:10.1145/3105909.
- 9 Davide Bresolin, Emilio Muñoz-Velasco, and Guido Sciavicco. On sub-propositional fragments of modal logic. *Logical Methods in Computer Science*, 14(2), 2018. doi:10.23638/LMCS-14(2:16)2018.
- 10 Marcelo Finger and Mark Reynolds. Two-dimensional executable temporal logic for bitemporal databases. In Howard Barringer, Michael Fisher, Dov Gabbay, and Graham Gough, editors, *Advances in Temporal Logic*, pages 393–411. Springer, 2000.
- 11 Dov M. Gabbay and Valentin B. Shehtman. Products of modal logics, part 1. *Logic Journal of the IGPL*, 6(1):73–146, 1998. doi:10.1093/JIGPAL/6.1.73.
- 12 Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991. doi:10.1145/115234.115351.
- 13 Miki Hermann and Gernot Salzer. Efficient learning of Horn formulas over finite totally ordered domains. In *Proc. of the 18th International Conference on Multi-disciplinary Trends in Artificial Intelligence*, volume 16354 of *Lecture Notes in Computer Science*, pages 80–92, 2026. doi:10.1007/978-981-95-4960-3_7.
- 14 Gerard Ligozat. Reasoning about cardinal directions. *Journal of Visual Languages and Computation*, 9(1):23–44, 1998. doi:10.1006/JVLC.1997.9999.
- 15 Carsten Lutz and Frank Wolter. Modal logics of topological relations. *Logical Methods in Computer Science*, 2(2), 2006. doi:10.2168/LMCS-2(2:5)2006.
- 16 Maarten Marx and Mark Reynolds. Undecidability of compass logic. *Journal of Logic and Computation*, 9(6):897–914, 1999. doi:10.1093/LOGCOM/9.6.897.
- 17 Angelo Montanari, Gabriele Puppis, and Pietro Sala. A decidable weakening of compass logic based on cone-shaped cardinal directions. *Logical Methods in Computer Science*, 11(4), 2015. doi:10.2168/LMCS-11(4:7)2015.
- 18 Yde Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990. doi:10.1305/NDJFL/1093635589.
- 19 Marc Vilain and Henry Kautz. Constraint propagation algorithms for temporal reasoning. In *Proc. of the 5th National Conference on Artificial Intelligence*, pages 377–382. Morgan Kaufmann, 1986. URL: <http://www.aaai.org/Library/AAAI/1986/aaai86-063.php>.