On the complexity of fragments of the modal logic of Allen's relations over dense structures

Publisher:
Springer Verlag

Published
DOI:10.1007/978-3-319-15579-1_40

On the Complexity of Fragments of the Modal Logic of Allen's Relations over Dense Structures

Davide Bresolin[1], Dario Della Monica[2], Angelo Montanari[3], Pietro Sala[4], and Guido Sciavicco[5]

[1] Department of Computer Science and Engineering
University of Bologna, Italy
davide.bresolin@unibo.it
[2] ICE-TCS, School of Computer Science
Reykjavik University, Iceland
dariodm@ru.is
[3] Department of Mathematics and Computer Science
University of Udine, Italy
angelo.montanari@uniud.it
[4] Department of Computer Science
University of Verona, Italy
pietro.sala@univr.it
[5] Department of Information, Engineering and Communications
University of Murcia, Spain
guido@um.es

Abstract. Interval temporal logics provide a natural framework for temporal reasoning about interval structures over linearly ordered domains, where intervals are taken as the primitive ontological entities. Their computational behaviour and expressive power mainly depend on two parameters: the set of modalities they feature and the linear orders over which they are interpreted. In this paper, we consider all fragments of Halpern and Shoham's interval temporal logic HS with a decidable satisfiability problem over the class of all dense linear orders, and we provide a complete classification of them in terms of their complexity and expressiveness by solving the last two open cases.

Keywords: computational complexity, interval temporal logics, satisfiability, expressiveness, decidability

1 Introduction

Most temporal logics proposed in the literature assume a point-based structure of time. They have been successfully applied in a variety of fields, ranging from the specification and verification of communication protocols to temporal data mining. However, a number of relevant application domains, such as, for instance, those of planning and synthesis of controllers, are often characterized by advanced features like durative actions (and their temporal relationships), accomplishments, and temporal aggregations, which are neglected or dealt with in an unsatisfactory way by point-based formalisms. The distinctive features of interval temporal logics turn out to be useful in these domains. As an example, they allow one to model telic statements [18], that is, statements that express goals or accomplishments, like the statement: "The airplane flew from Venice to Toronto" (see [8, Sect. II.B]). Temporal logics with interval-based semantics have also been proposed as suitable formalisms for the specification and verification of hardware [15] and of real-time systems [10]. Finally, successful implementations of interval-based systems can be found in the areas of learning (the adaptive learning system TERENCE [11], that provides a support to poor comprehenders and their educators, is based on the so-called Allen's interval algebra [3]) and real-time data systems (the algorithm RISMA [13], for performance and behaviour analysis of real-time data systems, is based on Halpern and Shoham's modal logic of Allen's relations [12]).

The variety of binary relations between intervals in a linear order was first studied by Allen [3], who investigated their use in systems for time management and planning. In [12], Halpern and Shoham introduced and systematically analyzed the (full) modal logic of Allen's relations (HS for short), that features one modality for each Allen relation. In particular, they showed that HS is highly undecidable over most classes of linear orders. This result motivated the search for (syntactic) fragments of HS offering a good balance between expressiveness and computational complexity. During the last decade, a systematic analysis has been carried out to characterize the complexity of the satisfiability problem for HS fragments [4, 5, 16], as well as their relative expressive power [1, 2, 5]. Such an analysis pointed out that such characterizations also depend on the class of linearly ordered set over which formulae are interpreted.

This paper aims at completing the classification of decidable HS fragments with respect to both their complexity and expressiveness, relative to the class of (all) dense linear orders. For our purposes, the class of dense linear orders and the linear order of the rational numbers $\mathbb{Q}$ are indistinguishable. Thus, all the results presented here directly apply to $\mathbb{Q}$ as well. The paper is organized as follows. In Section 2, we introduce syntax and semantics of (fragments of) HS. Next, in Section 3 we summarize known results about dense linear orders. In Section 4 and Section 5, we solve the last two open problems, thus completing the picture for the class of dense linear structures. It is worth mentioning that an analogous classification has been provided in [5] for the class of finite linear orders, the class of discrete linear orders, the linear order of the natural numbers $\mathbb{N}$, and the linear order of the integers $\mathbb{Z}$.

2 The Modal Logic of Allen's Relations

Let us consider a linearly ordered set $\mathbb{D} = \langle D, < \rangle$, where D is an element domain and $<$ is a total ordering on it. An *interval* over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in D$ and $x \leq y$. An interval is called a *point interval* if $x = y$ and a *strict interval* if $x < y$. In this paper, we assume the *strict semantics*, that is, we exclude point intervals and only consider strict intervals. The adoption of the strict semantics, excluding point intervals, instead of the *non-strict semantics*,

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

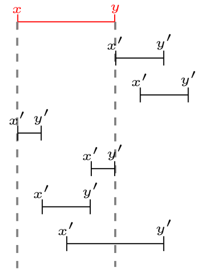


Fig. 1. Allen's interval relations and the corresponding HS modalities.

which includes them, conforms to the definition of interval adopted by Allen in [3], but differs from the one given by Halpern and Shoham in [12]. It has at least two strong motivations: first, a number of representation paradoxes arise when the non-strict semantics is adopted, due to the presence of point intervals, as pointed out in [3]; second, when point intervals are included there seems to be no intuitive semantics for interval relations that makes them both pairwise disjoint and jointly exhaustive. If we exclude the identity relation, there are 12 different relations between two strict intervals in a linear order, often called *Allen's relations* [3]: the six relations R_A (*meets* or *adjacent*), R_L (*after* or *later*), R_B (*starts* or *begins*), R_E (*finishes* or *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 modality $\langle X \rangle$ with each Allen relation R_X. For each $X \in \{A, L, B, E, D, O\}$, the *transpose* of modality $\langle X \rangle$ is modality $\langle \overline{X} \rangle$, corresponding to the inverse relation $R_{\overline{X}}$ of R_X. Halpern and Shoham's logic HS [12] is a multi-modal logic with formulae built from a finite, non-empty set $\mathcal{AP}$ of atomic propositions (also referred to as proposition letters), the propositional connectives $\vee$ and $\neg$, and a modality for each Allen relation. With every subset $\{R_{X_1}, \ldots, R_{X_k}\}$ of these relations, we associate the fragment $\mathsf{X_1 X_2 \ldots X_k}$ of HS, whose formulae are defined by the grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X_1 \rangle \varphi \mid \ldots \mid \langle X_k \rangle \varphi,$$

where $p \in \mathcal{AP}$. The other propositional connectives and constants (e.g., $\wedge$, $\rightarrow$, 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.

The (strict) semantics of HS 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* $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}(\mathbb{D})}$, 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 on a given interval $[x, y]$ in an interval model M is defined by structural induction on formulae as follows:

– $M, [x, y] \Vdash p$ if and only if $[x, y] \in V(p)$, for each $p \in \mathcal{AP}$;

– $M, [x, y] \Vdash \neg\psi$ if and only if it is not the case that $M, [x, y] \Vdash \psi$;
– $M, [x, y] \Vdash \varphi \vee \psi$ if and only if $M, [x, y] \Vdash \varphi$ or $M, [x, y] \Vdash \psi$;
– $M, [x, y] \Vdash \langle X \rangle \psi$ if and only if there exists $[x', y']$ such that $[x, y] R_X [x', y']$ and $M, [x', y'] \Vdash \psi$, for each modality $\langle X \rangle$.

Formulae of HS can be interpreted over a given class of interval models; we identify the class of interval models over linear orders in $\mathcal{C}$ with the class $\mathcal{C}$ itself. Thus, we will use, for example, the expression 'formulae of HS are interpreted over the class $\mathcal{C}$ of linear orders' instead of the extended one 'formulae of HS are interpreted over the class of interval models over linear orders in $\mathcal{C}$'. Among others, we mention the following important classes of linear orders: *(i)* the class of *all* linear orders Lin; *(ii)* the class of all *dense* linear orders Den, that is, those in which for every pair of different points there exists at least one point in between them; *(iii)* the class of all *weakly discrete* linear orders WDis, that is, those in which every element, apart from the greatest one, if it exists, has an immediate successor, and every element, other than the least one, if it exists, has an immediate predecessor; *(iv)* the class of all *strongly discrete* linear orders Dis, that is, those in which for every pair of different points there are only finitely many points in between them; *(v)* the class of all *finite* linear orders Fin, that is, those having only finitely many points; *(vi)* the singleton classes consisting of the standard linear orders over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, and $\mathbb{N}$. The *mirror image* (or, simply, *mirror*) of a fragment $\mathcal{F}$ is obtained by simultaneously substituting $\langle A \rangle$ with $\langle \overline{A} \rangle$, $\langle B \rangle$ with $\langle E \rangle$, $\langle \overline{B} \rangle$ with $\langle \overline{E} \rangle$, $\langle O \rangle$ with $\langle \overline{O} \rangle$, $\langle L \rangle$ with $\langle \overline{L} \rangle$, and the other way around. When interpreted over left/right symmetric classes of structures (i.e., classes $\mathcal{C}$ such that if $\mathcal{C}$ contains a linear order $\mathbb{D} = \langle D, \prec \rangle$, then it also contains a linear order isomorphic to its dual linear order $\mathbb{D}^d = \langle D, \succ \rangle$, where $\succ$ is the inverse of $\prec$), such as Den, all computational properties of a fragment are preserved for its mirror one; thanks to this observation, we can safely deal with only one fragment for each pair of mirror fragments.

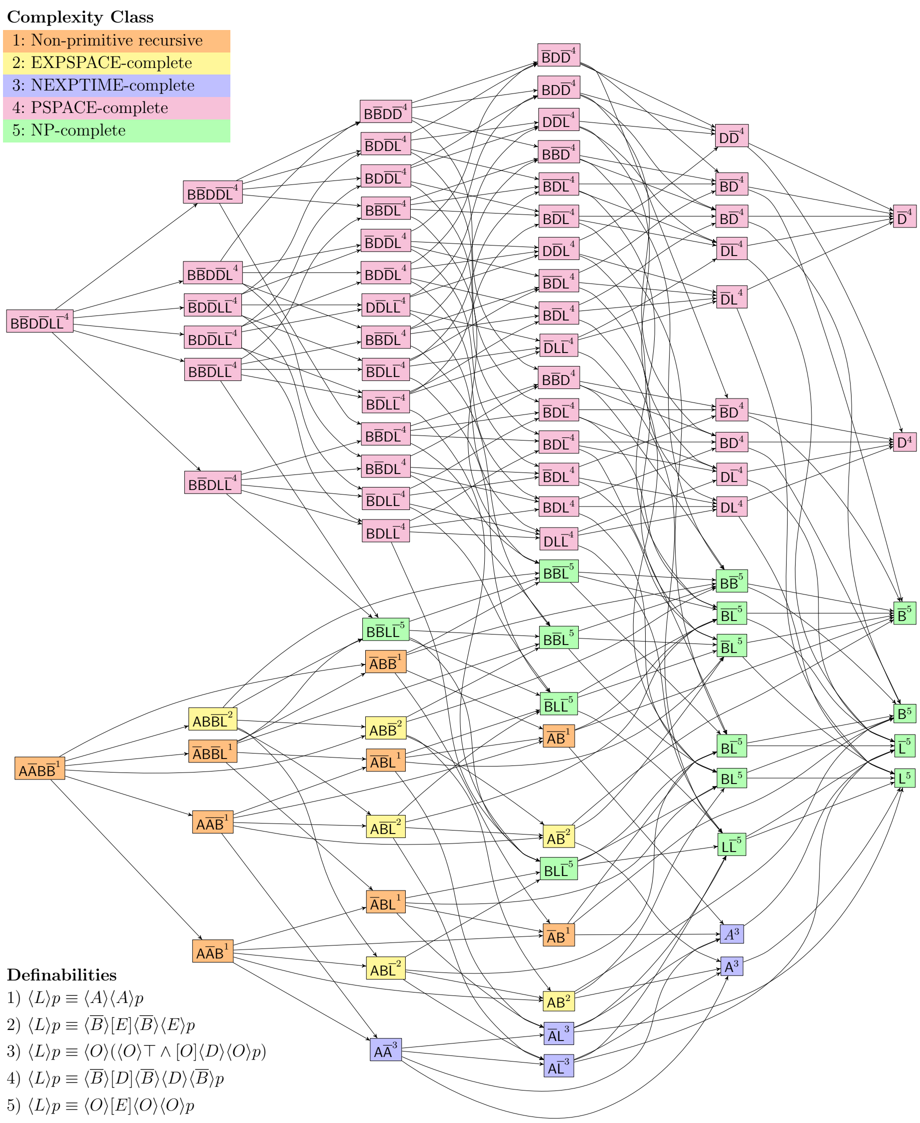
3 Known and Unknown Results

It has been proved in [1] that there are precisely 9 different optimal definabilities that hold among HS modalities in the dense case. As a consequence, only 966 HS fragments are expressively different (out of 4096 different subsets of 12 modalities). Of those, 146 are decidable, thanks to the following results:

Undecidability: we know from [4] that each fragment containing (as definable) O, AD, or $\mathsf{A\overline{D}}$ is undecidable;

Non-primitive recursive: the decidability of $\mathsf{A\overline{A}B\overline{B}}$ has been proved in [14], where it has also been shown that each fragment containing $\mathsf{A\overline{A}B}$ or $\mathsf{A\overline{A}\overline{B}}$ is non-primitive recursive;

ExpSpace-completeness: as a consequence of the results presented in [8], we know that $\mathsf{AB\overline{B}\overline{L}}$ is in ExpSpace, and each fragment containing AB or $\mathsf{A\overline{B}}$ is ExpSpace-hard (in particular, the hardness result given in [8] for $\mathsf{AB\overline{B}}$ can be suitably rephrased to deal with the smaller fragments AB and $\mathsf{A\overline{B}}$);



Definabilities

1) $\langle L\rangle p \equiv \langle A\rangle\langle A\rangle p$
2) $\langle L\rangle p \equiv \langle \overline{B}\rangle[E]\langle \overline{B}\rangle\langle E\rangle p$
3) $\langle L\rangle p \equiv \langle O\rangle(\langle O\rangle\top \wedge [O]\langle D\rangle\langle O\rangle p)$
4) $\langle L\rangle p \equiv \langle \overline{B}\rangle[D]\langle \overline{B}\rangle\langle D\rangle\langle \overline{B}\rangle p$
5) $\langle L\rangle p \equiv \langle O\rangle[E]\langle O\rangle\langle O\rangle p$
6) $\langle L\rangle p \equiv \langle O\rangle(\langle O\rangle\top \wedge [O]\langle B\rangle\langle O\rangle\langle O\rangle p)$
7) $\langle L\rangle p \equiv \langle O\rangle(\langle O\rangle\top \wedge [O][\overline{L}]\langle O\rangle\langle O\rangle p)$
8) $\langle O\rangle p \equiv \langle E\rangle\langle \overline{B}\rangle p$
9) $\langle D\rangle p \equiv \langle E\rangle\langle B\rangle p$

Fig. 2. Decidable fragments of HS in the dense case and their relative expressive power.

NExpTime-completeness: it has been proved in [7] that $\mathsf{A}\overline{\mathsf{A}}$ is in NExpTime, and both A and $\overline{\mathsf{A}}$ are NExpTime-hard;

PSpace-completeness: each sub-fragment of $\mathsf{B}\overline{\mathsf{B}}\mathsf{D}\overline{\mathsf{D}}\mathsf{L}\overline{\mathsf{L}}$ that contains (as definable) D or $\overline{\mathsf{D}}$ is shown to be PSpace-complete in [6, 16].

The purpose of this paper is to fill in the few gaps still uncovered by this collection of results. Here, we shall prove that: *(i)* $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$ and all its fragments are NP-complete (observe that each fragment is NP-hard, given that it is at least as expressive as propositional logic), and *(ii)* all the fragments that contain $\overline{\mathsf{A}}\mathsf{B}$ or $\overline{\mathsf{A}}\mathsf{B}$ are non-primitive recursive. The aforementioned results allow us to draw a picture that encompasses all HS fragments, ordered according to their relative expressive power and grouped by computational complexity. We show here such a picture (see Fig. 2), limited to all and only decidable HS fragments (for the sake of readability, we omit fragments that are expressively equivalent or mirror image of another fragment featured in the picture). In Fig. 2 we also show the 9 definabilities that hold among HS modalities over dense linear orders.

4 NP-Complete Fragments

In this section we show that the fragment $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$ is in NP (NP-completeness immediately follows as propositional logic is embedded into $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$). By defining a suitable notion of pseudo-model for formulae of $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$ we can show that each satisfiable formula admits a pseudo-model of size at most $P(|\varphi|)$ for some polynomial P. For lack of space, in this paper we only give the intuition behind the concept of pseudo-model and the main ideas behind the small pseudo-model theorem. A detailed account of the proof can be found in [9].

We start the discussion by considering the fragment $\mathsf{L}\overline{\mathsf{L}}$. The semantics of the interval modalities implies that intervals with the same ending point agree on the truth of $\langle L\rangle$-formulae (i.e., formulae of the kind $\langle L\rangle\varphi$); symmetrically, intervals with the same beginning point agree on $\langle\overline{L}\rangle$-formulae. Hence, given a model M for a formula φ, we can associate to every point x the set of its $\mathsf{L}\overline{\mathsf{L}}$-*requests*, defined as the pair of sets $(L_x, \overline{L}_x)$, where L_x contains all formulae ψ in the *closure of* φ (that is, the set of all sub-formulae of φ and their negations) such that $\langle L\rangle\psi$ is true over all intervals $[y, x]$, and $\overline{L}_x$ contains all formulae ψ in the closure of φ such that $\langle\overline{L}\rangle\psi$ is true over all intervals $[x, y]$. Since the closure of a formula is a finite set, we can partition the domain of the model into a finite number of clusters of points with the same set of $\mathsf{L}\overline{\mathsf{L}}$-requests. Moreover, by the transitivity of both $\langle L\rangle$ and $\langle\overline{L}\rangle$, we have that the set of $\mathsf{L}\overline{\mathsf{L}}$-requests is monotone with respect to the ordering of points, that is, for every pair of points $x < y$ we have $L_x \supseteq L_y$ and $\overline{L}_x \subseteq \overline{L}_y$. This implies that every cluster is either a single *point* or a *segment* of D, and that the number of clusters is at most $4|\varphi|$.

A pseudo-model for $\mathsf{L}\overline{\mathsf{L}}$ is an abstract representation of the partitioning. It is formally defined as a finite $\mathsf{L}\overline{\mathsf{L}}$-*sequence* of triples:

$$(L_0, \overline{L}_0, Type_0), (L_1, \overline{L}_1, Type_1), \ldots, (L_n, \overline{L}_n, Type_n),$$

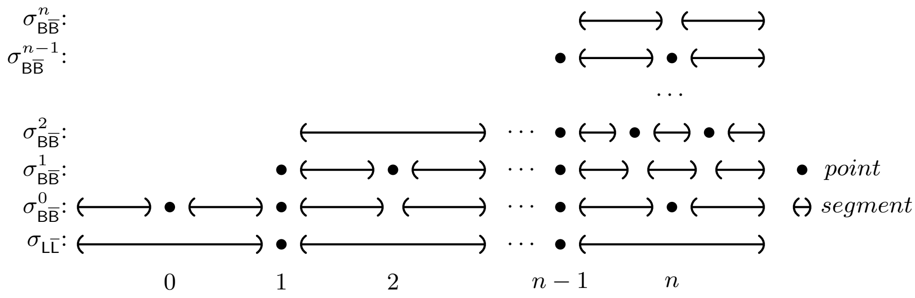


Fig. 3. A pseudo-model for $\mathsf{B\overline{B}L\overline{L}}$.

where each *Type* is either *point* or *segment*, and such that: *(i)* the monotonicity of $\mathsf{L\overline{L}}$-requests is respected; *(ii)* the first and the last triple of the sequence are of type *segment*; *(iii)* clusters of type *point* cannot be adjacent. To represent a well-formed model for φ, an $\mathsf{L\overline{L}}$-sequence must respect the following additional constraints:

- it must be *consistent*: for every pair of indexes $i < j$ there must exists an *atom* F (that is, a maximally consistent subset of the closure) that contains the formula $\langle L\rangle\psi$ for every $\psi \in L_j$, the formula $\neg\xi$ for every $\xi \notin L_i$, the formula $\langle\overline{L}\rangle\eta$ for every $\eta \in \overline{L}_i$ and the formula $\neg\zeta$ for every $\zeta \notin \overline{L}_j$;
- it must be L-*fulfilling*: for every index i and every formula $\psi \in L_i$ there must exists a pair of indexes $i < j < k$ and atom F containing ψ and consistent with the clusters j and k;
- it must be $\overline{\mathsf{L}}$-*fulfilling*, which is defined analogously.

The consistency condition guarantees that $[L]$- and $[\overline{L}]$-formulae are satisfied, while the fulfillment conditions guarantee that $\langle L\rangle$- and $\langle\overline{L}\rangle$-formulae are satisfied as well. We have already observed that the number of clusters (and thus, the length of an $\mathsf{L\overline{L}}$-sequence) is bounded by $4|\varphi|$. Hence, by guessing a $\mathsf{L\overline{L}}$-sequence and then checking it for consistency and fulfillment we can easily obtain an NP procedure for deciding the satisfiability of a formula in $\mathsf{L\overline{L}}$.

The extension of the above result to the full $\mathsf{B\overline{B}L\overline{L}}$ language is based on the following observation. Given a model for the formula and an interval $[x, y]$ we define the set of $\mathsf{B\overline{B}}$-*requests* of the interval as the pair $(B_{[x,y]}, \overline{B}_{[x,y]})$, where $B_{[x,y]}$ contains all formulae ψ in the closure of φ such that $\langle B\rangle\psi$ is true on $[x, y]$, and $\overline{B}_{[x,y]}$ contains all formulae ξ in the closure of φ such that $\langle\overline{B}\rangle\xi$ is true on $[x, y]$. Fixed a point x in the model, we have that the sets of $\mathsf{B\overline{B}}$-requests of the intervals $[x, y]$ with begin point x respect the same monotonicity property as for $\mathsf{L\overline{L}}$-requests: for every pair of points $y < z$ we have $B_{[x,y]} \subseteq B_{[x,z]}$ and $\overline{B}_{[x,y]} \supseteq \overline{B}_{[x,z]}$. Hence, it is possible to partition the intervals starting in any given point x into at most $4|\varphi|$ "points" and "segments". A pseudo-model for $\mathsf{B\overline{B}L\overline{L}}$ is then made of the following components (see Fig. 3 for a graphical account):

- an $\mathsf{L\overline{L}}$-sequence $\sigma_{\mathsf{L\overline{L}}} = (L_0, \overline{L}_0, Type_0), (L_1, \overline{L}_1, Type_1), \ldots, (L_n, \overline{L}_n, Type_n)$ defining the partitioning of $\mathsf{L\overline{L}}$-requests;

– for every cluster $(L_i, \overline{L}_i, Type_i)$ of the $\mathsf{L}\overline{\mathsf{L}}$-sequence, a $\mathsf{B}\overline{\mathsf{B}}$-*sequence* $\sigma^i_{\mathsf{B}\overline{\mathsf{B}}} = (B_i, \overline{B}_i, Type_i), (B_{i+1}, \overline{B}_{i+1}, Type_{i+1}), \ldots, (B_m, \overline{B}_m, Type_m)$ representing all intervals $[x, y]$ such that x belongs to the ith cluster $(L_i, \overline{L}_i, Type_i)$. $\sigma^i_{\mathsf{B}\overline{\mathsf{B}}}$ must be a refinement of the partitioning $(L_i, \overline{L}_i, Type_i) \ldots (L_n, \overline{L}_n, Type_n)$.

The consistency and the fulfillment condition are suitably extended to guarantee satisfiability of $\mathsf{B}\overline{\mathsf{B}}$-formulae. Since the size of a $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$ pseudo-model is quadratic in the size of the formula, we can easily obtain an NP decision procedure that guesses a pseudo-model and checks the satisfiability of a formula in $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$.

Theorem 1. *The satisfiability problem for the logic* $\mathsf{B}\overline{\mathsf{B}}\mathsf{L}\overline{\mathsf{L}}$ *and each one of its fragments, interpreted over the class of dense linear orders, is* NP*-complete.*

5 Non-Primitive Recursive Fragments

As we have mentioned, the last piece needed to complete the picture in Fig. 2 concerns the non-primitive recursive fragments. In [14] the non-primitive recursiveness of $\mathsf{A}\overline{\mathsf{A}}\mathsf{B}$ and $\mathsf{A}\overline{\mathsf{A}}\overline{\mathsf{B}}$ has been proved. We shall prove here that, in actuality, every fragment that contains $\overline{\mathsf{A}}\overline{\mathsf{B}}$ or $\overline{\mathsf{A}}\mathsf{B}$ is non-primitive recursive.

Lossy counter machines are a variant of Minsky counter automata where transitions may non-deterministically decrease the values of counters. A comprehensive survey on faulty machines and on the relevant complexity, decidability, and undecidability results can be found in [17]. Formally, a *counter automaton* is a tuple $\mathcal{A} = (Q, q_0, C, \Delta)$, where Q is a finite set of control *states*, $q_0 \in Q$ is the initial state, $C = \{c_1, \ldots, c_k\}$ is the set of *counters*, whose values range over $\mathbb{N}$, and Δ is a *transition relation*. The relation Δ is a subset of $Q \times L \times Q$, where L is the *instruction set* $L = \{inc, dec, ifz\} \times \{1, \ldots, k\}$. A *configuration* of $\mathcal{A}$ is a pair $(q, \bar{v})$, where $q \in Q$ and $\bar{v}$ is the vector of counter values. A *run* of a Minsky (i.e., with no error) counter automaton is a finite or infinite sequence of configurations such that, for every pair of consecutive configurations $(q, \bar{v}), (q', \bar{v}')$, a *transition* $(q, \bar{v}) \xrightarrow{l} (q', \bar{v}')$ has been taken (for some $(q, l, q') \in \Delta$). The value of $\bar{v}'$ is obtained from the value of $\bar{v}$ by performing instruction l, where $l = (dec, i)$ requires $v_i > 0$ and $l = (ifz, i)$ requires $v_i = 0$. In lossy machines, which is the type in which we are interested, once a faulty transition has been taken, counter values may have been decreased nondeterministically before or after the execution of the exact transition by an arbitrary natural number. We use the notation $(q, \bar{v}) \xrightarrow{l}_{\dagger} (q', \bar{v}')$ to denote that there exist $\bar{v}_{\dagger}, \bar{v}'_{\dagger}$ such that $\bar{v} \geq \bar{v}_{\dagger}$, $(q, \bar{v}_{\dagger}) \xrightarrow{l} (q', \bar{v}'_{\dagger})$, and $\bar{v}'_{\dagger} \geq \bar{v}'$, where the ordering $\leq$ is defined component-wise in the obvious way. We are interested here in the *non-termination problem* for lossy machines, defined as the problem of deciding whether $\mathcal{A}$ has at least one infinite run starting with the *initial configuration* $(q_0, \bar{0})$. This problem is non-primitive recursive [17].

Lemma 2. *There exists a reduction from the non-termination problem for lossy counter machines to the satisfiability problem for* $\overline{\mathsf{A}}\mathsf{B}$ *over the class of all dense linear orders.*

Proof. Let $\mathcal{A} = (Q, q_0, C, \Delta)$ be a lossy counter machine. We write an $\overline{\mathsf{A}}\mathsf{B}$-formula $\varphi_{\mathcal{A}}$ which is satisfiable over a dense linear order if and only if $\mathcal{A}$ has at least one infinite run starting with the initial configuration. The computation is encoded left-to-right over a dense domain $\mathbb{D}$, by choosing an evaluation interval $[x, y]$ that works as the "last" one, and taking into account that, given any $x_0 < x$, there are infinitely many intervals between x_0 and x. We shall make use of the propositional letters u (*units*), q_i (*states*, where i ranges from 0 to $|Q|$), *conf* (*configurations*), c_i (*counters' instances*, where i ranges from 1 to $|C|$), and *corr*, $corr_i$ (*corresponds*; i ranges from 1 to $|C|$). Counters' instances, or simply *counters*, allow us to encode the counters of $\mathcal{A}$: given a configuration where the value of the i-th counter is n, the corresponding *conf*-interval will contain precisely n c_i-intervals. (By p-*interval* we denote those intervals that satisfy p, for every propositional letter p.) Additional propositional letters will be used in the reduction for technical reasons.

Let $[G]$ (*universal modality*) be the following shortcut:

$$[G]\varphi = \varphi \wedge [B]\varphi \wedge [\overline{A}]\varphi \wedge [\overline{A}][\overline{A}]\varphi.$$

The first step in our construction consists in *discretizing* the domain, making use of a propositional variable u. In doing so, we also set the first configuration:

$$\varphi_{u-chain} = \begin{cases} \langle\overline{A}\rangle\langle\overline{A}\rangle(u \wedge conf \wedge start \wedge q_0) \wedge [\overline{A}](\langle\overline{A}\rangle u \rightarrow \langle B\rangle u) \\ [G](u \rightarrow [B]\neg u) \wedge [G](u \rightarrow [B]u_b) \wedge [G](u \rightarrow [\overline{A}]\neg u_b) \\ [G](start \rightarrow u) \wedge [G](start \rightarrow [\overline{A}](\neg u \wedge [\overline{A}]\neg u)) \end{cases}$$

Consider an interval $[x, y]$ over which the formula of our reduction is evaluated. The sense of the above formula $\varphi_{u-chain}$ is to generate an infinite discrete chain $x_0, x_1, \ldots$ such that $x_0 < x_1 < \ldots < x < y$, and that each $[x_k, x_{k+1}]$ is labeled by u. With the above formulae we also guarantee that *start* is unique and no u-interval overlaps a u-interval in the chain.

With the next formulae we make sure that there is a infinite sequence of configurations. The first one (*start*) coincides with the unit $[x_0, x_1]$, and contains the starting state q_0 only. This is consistent with our requirement that all counters start with the value 0. Moreover, we guarantee that configurations' endpoints coincide with endpoints of elements of the u-chain, that every configuration contains a state, and that *start* is unique. In our reduction, the state is placed on the last unit of every configuration.

$$\varphi_{conf-chain} = \begin{cases} [G](conf \rightarrow (u \vee \langle B\rangle u)) \wedge [G](\langle\overline{A}\rangle conf \rightarrow \langle\overline{A}\rangle u) \\ [\overline{A}](\langle\overline{A}\rangle conf \rightarrow \langle B\rangle conf) \wedge [G](conf \rightarrow [B]conf_b \wedge [B]\neg conf) \\ [G](conf \rightarrow [\overline{A}]\neg conf_b) \wedge [G](\langle\overline{A}\rangle conf \leftrightarrow \langle\overline{A}\rangle(\bigvee_{i=0,\ldots,|Q|} q_i)) \end{cases}$$

Notice that states (q_i-intervals) occur exactly as last u-intervals of configurations. Since configurations do not overlap, this implies that each configuration contains exactly one state.

Configurations also contain counters' instances c_i for each counter i whose value is greater than zero. Besides, a special placeholder c_i^+ or c_i^- may be placed in a configuration, in order to make it possible to deal with increment and decrement operations. States, counters' instances, and placeholders may only hold over units, which, in turn, all have to contain one of the above. A placeholder must be placed over the counter to which it refers. Moreover, counters and states are mutually incompatible, and there cannot be more than one per type on a given unit. These requirements are guaranteed by the following formula:

$$\varphi_{units} = \begin{cases} [G](\bigwedge_{i=0,\ldots,|Q|}(q_i \rightarrow u) \wedge \bigwedge_{i=1,\ldots,|C|}((c_i \vee c_i^+ \vee c_i^-) \rightarrow u)) \\ [G](u \rightarrow ((\bigvee_{i=0,\ldots,|Q|} q_i) \vee (\bigvee_{i=1,\ldots,|C|} c_i))) \\ [G] \bigwedge_{i=0,\ldots,|Q|}(q_i \rightarrow (\bigwedge_{j=i+1,\ldots,|Q|} \neg q_j)) \\ [G] \bigwedge_{i=0,\ldots,|Q|}(q_i \rightarrow (\bigwedge_{j=1,\ldots,|C|} \neg c_j)) \\ [G] \bigwedge_{i=1,\ldots,|C|}((c_i \rightarrow (\bigwedge_{j=i+1,\ldots,|C|} \neg c_j)) \wedge (c_i^- \rightarrow c_i) \wedge (c_i^+ \rightarrow c_i)) \end{cases}$$

Before we can actually encode the transition function Δ, we have to axiomatize the properties of *corr* and $corr_i$ for each i. In a perfect (non-faulty) machine, when a counter is not modified by any operation from a configuration to the next one its value is preserved. Since we are encoding a lossy machine, it suffices to guarantee that no counter's value is ever incremented, except for the special case of an incrementing operation. To this end, we use the propositional letter *corr* as a basis for correspondence, and the proposition $corr_i$ to identify the correspondence for the i-th counter:

$$\varphi_{corr} = \begin{cases} [G] \bigwedge_{i=1,\ldots,|C|}(((c_i \wedge \neg c_i^+) \rightarrow \langle \overline{A} \rangle corr_i) \wedge (c_i^+ \rightarrow \neg \langle \overline{A} \rangle corr_i)) \\ [G] \bigwedge_{i=1,\ldots,|C|}(corr_i \rightarrow corr) \\ [G] \bigwedge_{i=1,\ldots,|C|}(corr_i \rightarrow \langle \overline{A} \rangle (c_i \wedge \neg c_i^-)) \wedge [G](corr \rightarrow [B] corr_b) \\ [G](((\bigvee_{i=0,\ldots,|Q|} q_i) \wedge corr_b) \rightarrow {corr_b}^*) \\ [G]((\bigvee_{i=0,\ldots,|Q|} q_i) \rightarrow [\overline{A}](corr_b \rightarrow {corr_b}^*)) \\ [G](corr \rightarrow [B] \neg corr) \wedge [G]({corr_b}^* \rightarrow [B] \neg {corr_b}^*) \\ [G](\langle \overline{A} \rangle {corr_b}^* \rightarrow \langle \overline{A} \rangle u) \wedge [G](corr \rightarrow \langle B \rangle {corr_b}^*) \\ [G]((u \wedge \neg (\bigvee_{i=0,\ldots,|Q|} q_i)) \rightarrow [\overline{A}] \neg {corr_b}^*) \end{cases}$$

To finalize the reduction, we now take care of incrementing and decrementing operations, as well as of the zero test. For each $(q, l, q') \in \Delta$, let $conf_{(q,l,q')}$ be a special propositional letter holding on a configuration and carrying information on which transition produced that configuration. Clearly, every configuration but *start* is the result of precisely one transition. Therefore, we have:

$$\varphi_{conf} = \begin{cases} [G]((conf \wedge \neg start) \leftrightarrow (\bigvee_{(q,l,q') \in \Delta} conf_{(q,l,q')})) \\ [G](\bigwedge_{(q,l,q') \in \Delta}(conf_{(q,l,q')} \rightarrow (\bigwedge_{(q'',l',q''') \neq (q,l,q')} \neg conf_{(q'',l',q''')}))) \end{cases}$$

We can now implement the actual transitions. To deal with the increment (resp., decrement) operation we make use of the symbol c_i^+ (resp., c_i^-), as follows:

$$\varphi_{inc} = \begin{cases} [G](\bigwedge_{(q,(inc,i),q')\in\Delta}(conf_{(q,(inc,i),q')} \rightarrow (\langle\overline{A}\rangle q \wedge \langle B\rangle c^+_{i,b}))) \\ [G](\bigwedge_{(q,(inc,i),q')\in\Delta}(\langle\overline{A}\rangle conf_{(q,(inc,i),q')} \rightarrow \langle\overline{A}\rangle q')) \\ [G](\bigwedge_{i=1,\ldots,|C|}(\langle\overline{A}\rangle c^+_{i,b} \leftrightarrow \langle\overline{A}\rangle c^+_i)) \\ [G](\bigwedge_{i=1,\ldots,|C|}(c^+_{i,b} \rightarrow (\langle\overline{A}\rangle conf \wedge [B]\neg conf))) \\ [G](\bigwedge_{i,j=1,\ldots,|C|}(c^+_{i,b} \rightarrow [B]\neg c^+_{j,b})) \\ [G](\bigwedge_{i=1,\ldots,|C|}((conf \wedge \langle B\rangle c^+_{i,b}) \rightarrow (\bigvee_{q,q'\in Q} conf_{(q,(inc,i),q')}))) \end{cases}$$

$$\varphi_{dec} = \begin{cases} [G](\bigwedge_{(q,(dec,i),q')\in\Delta}(conf_{(q,(dec,i),q')} \rightarrow (\langle\overline{A}\rangle q \wedge [\overline{A}](conf \rightarrow \langle B\rangle c^-_{i,b})))) \\ [G](\bigwedge_{(q,(dec,i),q')\in\Delta}(\langle\overline{A}\rangle conf_{(q,(dec,i),q')} \rightarrow \langle\overline{A}\rangle q')) \\ [G](\bigwedge_{i=1,\ldots,|C|}(\langle\overline{A}\rangle c^-_{i,b} \leftrightarrow \langle\overline{A}\rangle c^-_i)) \\ [G](\bigwedge_{i=1,\ldots,|C|}(c^-_{i,b} \rightarrow (\langle\overline{A}\rangle conf \wedge [B]\neg conf))) \\ [G](\bigwedge_{i,j=1,\ldots,|C|}(c^-_{i,b} \rightarrow [B]\neg c^-_{j,b})) \\ [G](\bigwedge_{i=1,\ldots,|C|}((conf \wedge \langle\overline{A}\rangle\langle B\rangle c^-_{i,b}) \rightarrow (\bigvee_{q,q'\in Q} conf_{(q,(dec,i),q')}))) \end{cases}$$

$$\varphi_{ifz} = \begin{cases} [G](\bigwedge_{(q,(ifz,i),q')\in\Delta}(conf_{(q,(ifz,i),q')} \rightarrow (\langle\overline{A}\rangle q \wedge [\overline{A}](conf \rightarrow [B]c^z_{i,b})))) \\ [G](\bigwedge_{(q,(ifz,i),q')\in\Delta}(\langle\overline{A}\rangle conf_{(q,(ifz,i),q')} \rightarrow \langle\overline{A}\rangle q')) \\ [G](\bigwedge_{i=1,\ldots,|C|}(((\langle\overline{A}\rangle c_i \rightarrow [\overline{A}]c^z_{i,b}) \wedge (\neg\langle\overline{A}\rangle c_i \rightarrow [\overline{A}]\neg c^z_{i,b}))) \end{cases}$$

The formula $\varphi_{u-chain} \wedge \varphi_{conf-chain} \wedge \varphi_{units} \wedge \varphi_{corr} \wedge \varphi_{conf} \wedge \varphi_{inc} \wedge \varphi_{dec} \wedge \varphi_{ifz}$ is satisfiable if and only if $\mathcal{A}$ has at least one infinite run. □

Since it is possible to construct a similar reduction using the fragment $\overline{\mathsf{AB}}$, we can conclude the following theorem.

Theorem 3. *The complexity of the satisfiability problem for the fragments* $\overline{\mathsf{AB}}$ *and* $\overline{\mathsf{A}}\mathsf{B}$ *over the class of dense linear orders is non-primitive recursive.*

6 Conclusions

In this paper, we solved the last open problems about the complexity of HS fragments whose satisfiability problem is decidable when interpreted over the class of dense linear orders (equivalently, $\mathbb{Q}$). If we look at the emerging picture, we notice that such a class turns out to be the best one from the point of view of computational complexity. The satisfiability problem for any HS fragment over the class of finite (resp, discrete) linear orders, as well as over $\mathbb{N}$ and $\mathbb{Z}$, is indeed at least as complex as over the class of dense linear orders. Moreover, there are some fragments, like the logic of subintervals D, for which the problem is decidable (in fact, PSPACE) over the latter class and undecidable over the former ones. The same relationships hold between the class of dense linear orders and the class of all linear orders (resp., $\mathbb{R}$) with respect to the known fragments.

Acknowledgements. The authors acknowledge the support from the Spanish fellowship program *'Ramon y Cajal' RYC-2011-07821* (G. Sciavicco), the project *Processes* *and Modal Logics* (project nr. 100048021) and the project *Decidability and Expressiveness for Interval Temporal Logics* (project nr. 130802-051) of the Icelandic Research Fund (D. Della Monica), and the Italian GNCS project *Automata, Games, and Temporal Logics for the verification and synthesis of controllers in safety-critical systems* (D. Bresolin, A. Montanari).

References

1. Aceto, L., Della Monica, D., Goranko, V., Ingólfsdóttir, A., Montanari, A., Sciavicco, G.: A complete classification of the expressiveness of interval logics of Allen's relations: the dense and the general case. Acta Informatica (2014), in press
2. Aceto, L., Della Monica, D., Ingólfsdóttir, A., Montanari, A., Sciavicco, G.: On the expressiveness of the interval logic of Allen's relations over finite and discrete linear orders. In: Proc. of the 14th JELIA, LNCS, vol. 8761, pp. 267 – 281. Springer (2014)
3. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
4. Bresolin, D., Della Monica, D., Goranko, V., Montanari, A., Sciavicco, G.: The dark side of interval temporal logic: marking the undecidability border. Annals of Mathematics and Artificial Intelligence 71(1-3), 41–83 (2014)
5. Bresolin, D., Della Monica, D., Montanari, A., Sala, P., Sciavicco, G.: Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. Theoretical Computer Science (2014), in press
6. Bresolin, D., Goranko, V., Montanari, A., Sala, P.: Tableaux for logics of subinterval structures over dense orderings. Journal of Logic and Computation 20(1), 133–166 (2010)
7. Bresolin, D., Montanari, A., Sala, P., Sciavicco, G.: Optimal tableau systems for propositional neighborhood logic over all, dense, and discrete linear orders. In: Proc. of the 20th TABLEAUX. LNAI, vol. 6793, pp. 73–87. Springer (2011)
8. Bresolin, D., Montanari, A., Sala, P., Sciavicco, G.: What's decidable about Halpern and Shoham's interval logic? The maximal fragment $\mathsf{AB\overline{B}\overline{L}}$. In: Proc. of the 26th LICS. pp. 387–396 (2011)
9. Bresolin, D., Della Monica, D., Montanari, A., Sala, P., Sciavicco, G.: On the complexity of fragments of the modal logic of Allen's relations over dense structures (extended version) (2014), available at `http://www.icetcs.ru.is/dario/techrep/lata15ext.pdf`
10. Chaochen, Z., Hansen, M.R.: Duration calculus: A formal approach to real-time systems. EATCS: Monographs in Theoretical Computer Science, Springer (2004)
11. Gennari, R., Tonelli, S., Vittorini, P.: An AI-based process for generating games from flat stories. In: Proc. of the 33th SGAI. pp. 337–350 (2013)
12. Halpern, J., Shoham, Y.: A propositional modal logic of time intervals. Journal of the ACM 38(4), 935–962 (1991)
13. Laban, S., El-Desouky, A.: RISMA: A rule-based interval state machine algorithm for alerts generation, performance analysis and monitoring real-time data processing. In: Proc. of the European Geosciences Union General Assembly. Geophysical Research Abstracts, vol. 15 (2013)
14. Montanari, A., Puppis, G., Sala, P.: Decidability of the interval temporal logic $\mathsf{A\overline{A}B\overline{B}}$ over the rationals. In: Proc. of the 39th MFCS. LNCS, vol. 8634, pp. 451–463. Springer (2014)

15. Moszkowski, B.: Reasoning about digital circuits. Tech. rep. stan-cs-83-970, Dept. of Computer Science, Stanford University, Stanford, CA (1983)
16. Sala, P.: Decidability of Interval Temporal Logics. Ph.D. thesis, Department of Mathematics and Computer Science, University of Udine, Udine, Italy (2010)
17. Schnoebelen, P.: Lossy counter machines decidability cheat sheet. In: Proc. of the 4th International Workshop (RP 2010). pp. 51–75 (2010)
18. Terenziani, P., Snodgrass, R.T.: Reconciling point-based and interval-based semantics in temporal relational databases: A treatment of the telic/atelic distinction. IEEE Transactions on Knowledge and Data Engineering 16(5), 540–551 (2004)