A New Modal Approach to the Logic of Intervals

ALTAF HUSSAIN, *Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ, UK.*
E-mail: ah701@doc.ic.ac.uk

Abstract

In Artificial Intelligence there is a need for reasoning about continuous processes, where assertions refer to time intervals rather than time points. Taking our lead from van Benthem's treatment of interval temporal structures and Halpern and Shoham's work on intervals, we present a interval temporal logic with two binary relations, precedence and inclusion. We study the logic in its full generality without making any assumptions about the underlying nature of time, be it discrete or dense, linear or branching. We identify two general classes of interval temporal structures, *minimal interval structures* and *van Benthem minimal interval structures*. We show that in our interval temporal language, the two classes in fact have the same logic. We go on to prove that the logic of minimal interval structures is complete and decidable, possessing the finite model property, and that the satisfiability problem is PSPACE-complete. In order to establish the complexity result we extend the tableau method introduced by Horrocks et al., which treats transitive and inverse relations only, to also incorporate interaction between relations. We go on to identify some important limitations in the expressive power of our logic, before concluding the paper by raising a number of interesting questions that follow from our work.

Keywords: Modal logic, tableau algorithm, interval logics, complexity.

1 Background and motivation

A point-based perspective of time has always had its philosophical detractors. Consider Zeno's paradox of the flying arrow, which, it is argued, cannot change position at an isolated moment of time and thus cannot move at all. It suggests that there is something problematic in the representation of time as a series of durationless moments if we want to describe the concept of continuous movement.

Such considerations resurface when we examine the formal treatment of time in theoretical computer science. Temporal logic was introduced by the philosopher Arthur Prior [34], it was first applied in theoretical computer science to reason about programs [33]. The initial perspective was point-based: formulas were interpreted over time points, and the temporal structures were typically assumed to be discrete. However, subsequent research in temporal logic began to concentrate on intervals rather than points. Again, it was argued that the interval-based representation of time was simpler and more natural in formalizing common sense reasoning than the standard scientific models, especially when reasoning about continuous processes [14]. Thus in computer science, work began on *process logic* [22, 31, 32], where intervals (or 'paths') represent pieces of computation, and *interval temporal logic* [20]. A concise review of these earlier interval logics is given in [24].

Vol. 17 No. 2,
doi:10.1093/logcom/exl007

Despite the early interest in interval-based temporal logics, it remains a significantly less-studied area of temporal logic compared to its point-based rival. No doubt, the 'negative results' obtained by Halpern and Shoham [24] have done much to temper the mood. Nonetheless, we strongly feel that an interval-based approach is an interesting perspective and, notwithstanding the 'negative results', there are many interesting points raised by the work of Halpern and Shoham that can be addressed positively. It is worth pointing out that we are *not* arguing that intervals are 'better' than points, but only that they deserve more attention than they have been afforded recently.

However, when we adopt an interval-based prespective, we have some very basic decisions to make about what are intervals, about what are the natural relations between intervals, and about any restrictions there should be on the valuation of atoms. In this article the choices we will make will be motivated by previous work on intervals; in particular it builds on both the philosophical and computer science perspective gained from the work of van Benthem [42] and Halpern and Shoham [24].

1.1 The logic of Halpern and Shoham

In [24], Halpern and Shoham present an interval logic *HS* which can be viewed as a generalization of point-based modal temporal logic. *HS* is a temporal logic with the following modalities: $\langle B \rangle$, $\langle E \rangle$, $\langle A \rangle$, $\langle \overline{B} \rangle$, $\langle \overline{E} \rangle$, $\langle \overline{A} \rangle$, which have the following intended readings:

$\langle B \rangle\phi$ ϕ holds at a strict beginning interval of the current one
$\langle B \rangle\phi$ ϕ holds at a strict end interval of the current one
$\langle B \rangle\phi$ ϕ holds at an interval met by the current one, i.e. it begins where the current one ends
$\langle \overline{B} \rangle\phi$ ϕ holds at an interval which has the current one as a beginning interval
$\langle \overline{E} \rangle\phi$ ϕ holds at an interval which has the current one as a ending interval
$\langle \overline{A} \rangle\phi$ ϕ holds at an interval meeting the current one

For the semantics, they opt for temporal structures $(T, \leq)$ where T is a set of points and $\leq$ is a partial order on T. Intervals are then defined as (convex) sets of points. Their choice of modalities suffice to capture the 13 possible relations between distinct intervals in a linear temporal structure, as is illustrated in Figure 1. They go on to show that for most interesting classes of temporal structures, validity and satisfiability is undecidable. One of the results they establish states the following:

Fact 1.1
The validity problem for each of the following classes of temporal structures is r.e.-complete (a problem X is r.e.-complete if a Turing machine can list all the 'yes' instances of the problem, and any other r.e. problem can be reduced to X by a Turing machine):

1. the class of all temporal structures,
2. the class of all linear temporal structures,
3. the class of all discrete temporal structures,
4. the class of all dense temporal structures,
5. the class of all dense, linear, unbounded temporal structures.

A *complete* temporal structure is one in which any sequence with an upper bound has a least upper bound; a class of temporal structures is said to be complete if *all* structures in the class are complete [24, p. 957]. For classes that are complete as well as containing an infinitely

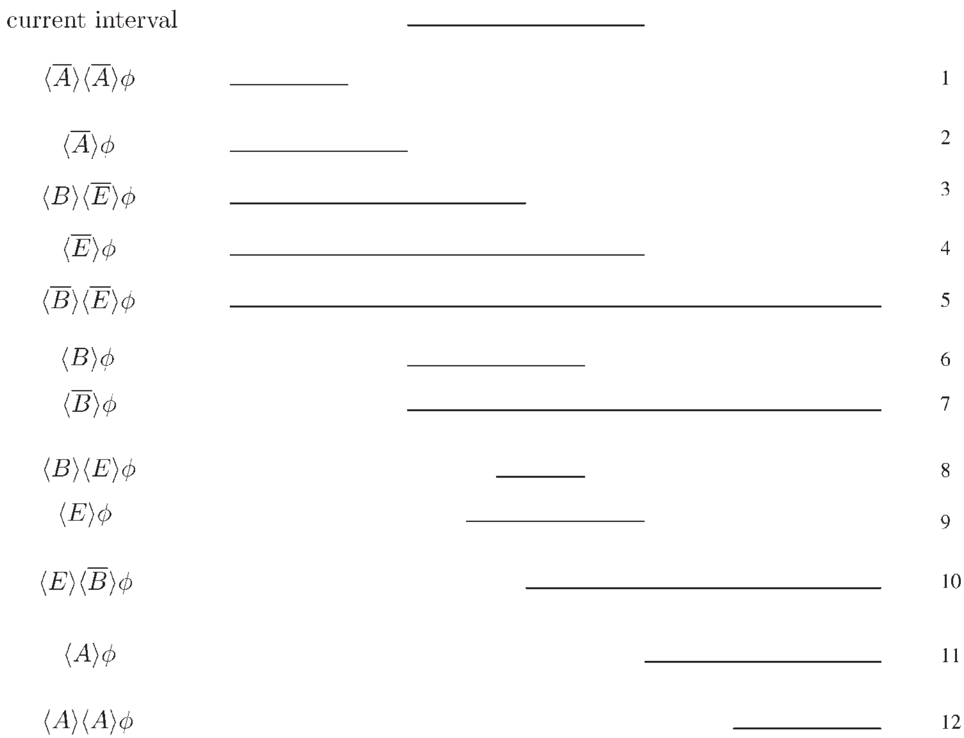

FIGURE 1. This figure illustrates the different relations that hold between any two intervals. For example, item 1 depicts an interval (at which ϕ holds) which precedes the current interval. Therefore at the current interval $\langle\overline{A}\rangle\langle\overline{A}\rangle\phi$ holds with respect to the interval at 1.

ascending sequence, they show that the validity problem is even harder. One of the interesting open problems they raise at the end of the paper asks: *what happens to the complexity of the validity problem if we modify the logic, in particular what happens for weaker or incomparable combinations of modal operators*?

1.2 The logic of van Benthem

In [42], van Benthem gives a more abstract treatment of intervals. He does this by (1) combining several Allen relations [2] into just two relations; and (2) by passing from sets of points to abstract objects. He introduces the following structure for talking about intervals:

DEFINITION 1.2
A period structure $\mathcal{F}$ is an ordered triple $\langle I, \sqsubseteq, <\rangle$ of a non-empty set I carrying two binary relations $\sqsubseteq$ ('inclusion') and $<$ ('precedence').

As we have seen in the case of Halpern and Shoham, intervals are built over temporal structures $(T, \leq$, where T is a set of points and $\leq$ is a partial ordering. As a consequence of intervals being a set of points of a partial order, the precedence relation between intervals will be transitive; similarly, inclusion between intervals will also be transitive. Furthermore, the following conditions will hold between intervals x, y, z:

$\forall xy(x < y \rightarrow \forall u(u \sqsubseteq x \rightarrow u < y))$ (Right Monotonicity)
$\forall xy(x < y \rightarrow \forall u(u \sqsubseteq y \rightarrow x < u))$ (Left Monotonicity)

These properties are basic in the sense that they arise from sets of points of a partial order. It is therefore natural to adopt them as axioms in the abstract setting. What additional properties we should adopt is motivated by the choices van Benthem makes.

Whilst van Benthem's treatment of intervals is more abstract, his choices, like Halpern and Shoham's, are informed by concrete examples; in his case the choice of the minimal basis for period structures is informed by his investigation of intervals over $\mathbb{Z}$ and $\mathbb{Q}$.

He proposes that $<$ be a strict ordering, and $\sqsubseteq$ be a partial ordering. However, because his intention is to axiomatise intervals over $\mathbb{Z}$ and $\mathbb{Q}$, he also insists on the following condition:

$\forall xy(xOy \rightarrow \exists z(z \sqsubseteq x \wedge z \sqsubseteq y \wedge \forall u((u \sqsubseteq x \wedge u \sqsubseteq y) \rightarrow u \sqsubseteq z))$ (Conjectivity),
where $xOy =_{df} \exists z(z \sqsubseteq x \wedge z \sqsubseteq y)$.

This condition states that any two overlapping intervals have a greatest common subinterval. A greatest common subinterval will also be referred to as a *maximal* subinterval. It is questionable whether this property should be adopted as a basic property of interval structures, since it is not always valid if intervals are sets of points of a partial order.

A period structure in which $<$ is a strict ordering, $\sqsubseteq$ is a partial ordering, and monotonicity (left and right) and conjectivity are satisfied will be called a *van Benthem minimal interval structure*. We also identify a simpler class, which we call *minimal interval structures*, in which the conjectivity condition is dropped. While van Benthem studied some instances of these two classes, the Halpern and Shoham question of whether complexity of the logic drops if we pass to a more general setting remained open.

1.3 Initial choices for our logic

In attempting to give a positive answer to the Halpern and Shoham question, a good strategy for obtaining a simpler logic, in complexity terms, is to abstract and have less properties. As we have seen, the work of van Benthem provides clear guidelines on how to go about achieving this.

In this article, it is our intention to study the logic of intervals in its full generality, and we hope to show that the choices we make give rise to an interval logic with a simple syntax and semantics and which has a good computational complexity.

Ontology. Are intervals primitive objects in the logic, or are they defined in terms of points? In philosophy, you find logics of both kind. In computer science, almost all interval-based logics construct intervals from points (with Allen's logic [2] and the Event Calculus [27] being the only exceptions we are aware of). Because intervals as derived objects have been extensively studied in computer science, and because we believe treating intervals as first class citizens is worthy of consideration, we will join the minority by taking intervals as primitive objects.

Commitment to an underlying temporal structure. Most interval-based temporal logics in computer science have been committed to the discrete and linear view of time. Our logic (like the *HS* logic) will be quite general in this respect: we study two general classes of interval temporal structures, *minimal interval structures* and *van Benthem minimal interval structures*. By imposing only the most elementary constraints on our logic, we will not exclude branching and linear time, dense and discrete time, bounded and unbounded time, and so on.

Choice of operators. The strong commitment to a discrete and linear order, in computer science, dictated fairly standard modal operators. In philosophy, there has been less uniformity. Following van Benthem [42], we employ two very natural pairs of modal operators, precedence and inclusion for talking about interval temporal structures.

Evaluating formulas. An issue that arises when evaluating propositions in computer science is whether or not *locality* is assumed; a logic is local if a propositonal atom is true over an interval iff it is true over its starting point. In philosophy, an assumption sometimes made is that of *homogeneity*. A logic is homogeneous when, roughly speaking, a proposition is true over an interval iff it is true over all its subintervals. By taking intervals as primitives, we do not assume either homogeneity or locality.

1.4 Minimal interval structures as products of modal logics

Given the generality of our logic, we can view our approach from the vista of the modal logic of products. Introduced in the 1970s [38, 39], products of modal logics have been used for a variety of applications in mathematical logic, computer science and artificial intelligence as a multi-dimensional formalism (see, e.g. [7, 13, 15, 36, 46]). They have been studied extensively over the last decade, and many results have been obtained for them. For a comprehensive exposition and further references see [16]. Of particular interest to us, is a result in [17], which establishes that the products of *K*4 and *S*4 are undecidable. The problem arises from the two interaction axioms: Commutativity and Church-Rosser. The moral seems to be that if we want to study product logics which have good computational complexity, we need to weaken the interaction between the relations. Consequently, a general research problem can be formulated as follows: *is it possible to reduce the computational complexity of product logics by relaxing the interaction between their components and yet keeping some of the useful and attractive features of the product construction*? [18]. In our logic, we have in effect, a *K*4 modality ($<$) and an S4 modality ($\sqsubseteq$). However, the interaction between the modalities is weaker than that required between products. And this allows us to show that the logic of minimal interval structures is decidable. Intervals, therefore, provide a natural candidate for studying weaker versions of product logics.

1.5 The main contributions of the study

In this article, we establish the following results concerning our two classes of interval temporal structures:

1. Soundness and Completeness of an axiomatic system for both the class of minimal interval structures and the class of van Benthem minimal interval structures,
2. By the use of the bulldozing technique and a step-by-step construction we show that the class of minimal interval structures is the *same* as the class of van Benthem minimal interval structures,
3. We introduce the notion of a *saturated set* and again appeal to the bulldozing and step-by-step construction in order to prove the decidability of the logic of both classes,
4. We present a tableau algorithm in order to establish the PSPACE-completeness of the satisfiability problem for the logic of both classes. The proof builds on and extends the

tableau method presented in [25]. It does so by permitting disparate relations to interact, as engendered by the monotonicity condition in our logic. The complications that arise from this additional condition lead to a significant increase in the polynomial bound. In [25], the complexity of the satisfiability problem for the description logic $ALCNI_{R^+}$ (with transitive and converse relations) is shown to be $O(m^{10})$; the complexity of our logic is shown to be $O(m^{16})$. The detailed analysis of this complexity bound constitutes the main technical contribution of our article.

The decidability and relatively good computational complexity of our logic can be attributed to two main differences between our approach and that of Halpern and Shoham:

1. Our notion of what is an interval is weaker than theirs.
2. Our choice of modalities is expressively weaker than theirs.

Halpem and Shoham's notion of an interval is very different from ours. They build intervals over a (convex) set of points; they further restrict themselves to considering only *linear intervals*, which means that for any two points t_1 and t_2 such that $t_1 \leq t_2$, the set of points $\{t : t_1 \leq t \leq t_2\}$ is totally ordered; and they insist that the set of intervals is closed under certain operations, e.g., if $\langle x, y \rangle$ is an interval, then $\langle x, x \rangle$ is also an interval. By taking intervals as primitives, we are not required to make any of the above suppositions. Thus, the classes of temporal structures that they investigate are less general than the ones we consider. And because our notion of an interval is weaker than theirs, our modalities are also weaker in their expressivity. In Section 7 we highlight the expressive limitations of our logic.

While it could be argued that minimal interval structures are too weak to be considered as 'real' interval temporal structures, nonetheless, any 'real' interval temporal structure would satisfy the minimal constraints that we impose. As such we believe that it is worthwhile and instructive to begin our investigation of interval temporal structures by considering this general class. The question of what, if any, further assumptions are needed to obtain 'real' interval temporal structures we leave for future work.

Structure of the article. In the next section, we begin by introducing the class of minimal interval structures and also the class of pre-interval structures, which will play an important technical role throughout the article. We then define the interval temporal language we will use to talk about interval structures. In Section 3, we show that the logic of minimal interval structures is (strongly) sound and complete. The completeness proof is obtained by developing a general bulldozing technique that handles the interaction of the precedence and inclusion relations, and establishes that every pre-interval structure is the bounded morphic image of a minimal interval structure. The same technique can also be used to show that every van Benthem pre-interval structure is a bounded morphic image of a van Benthem minimal interval structure. In Section 4, we show that the logic of minimal interval structures is decidable. This is obtained by proving a general truth lemma that establishes, for any formula ϕ, if ϕ is satisfiable in a model based on a minimal interval structure, then ϕ is satisfiable in a finite model based on a pre-interval structure. In Section 5, we establish the decidability of the logic of van Benthem minimal interval structures by showing that the logic is the *same* as the logic of minimal interval structures. In Section 6, we present a tableau algorithm to show that the satisfiability problem for the logic of minimal interval structures is PSPACE-complete. We do so by extending the tableau method presented in [25] to handle the complications that arise from the interaction between the relations. In Section 6, we highlight some important expressive limitations of our logic. In Section 7, we conclude the article.

2 The modal logic of intervals

In this section we introduce the class of minimal interval structures and our *interval temporal language* for talking about them.

Definition 2.1
A frame $\mathcal{F} = \langle W, <, >, \sqsubseteq, \sqsupseteq \rangle$ is a **minimal interval structure** if it satisfies the following conditions: for $<$ (and its converse, $>$) *Irreflexivity* and *Transitivity*; for $\sqsubseteq$ (and its converse, $\sqsupseteq$) *Reflexivity*, *Transitivity* and *Antisymmetry*, plus the following interaction axioms: *Right Monotonicity* and *Left Monotonicity*.

As van Benthem observed [42], interval temporal structures can be built up by taking any subset of the power set of a (point-based) strict partial ordering $(S, \prec)$, by giving suitable definitions of precedence and inclusion based on $\prec$.

If we take intervals as *derived* objects from an underlying set of points, then these conditions are preserved. In order to show this we will use *concrete frames* in which intervals are constructed from points to capture clearly the different notions of intervals and their relations over the respective structures. In what follows, let $T = (S, \prec)$. Then *concrete frames* consist of:

Definition 2.2
(Closed Frames (1)). Let $\mathrm{F} = (I(T), <, >, \sqsubseteq, \sqsupseteq)$, where $I(T)$ consists of all non-empty closed intervals $[a, b]$, and $[a, b] < [c, d]$ if $b < c$ ($>$ is defined as the converse of $<$), and $[a, b] \sqsubseteq [c, d]$ if $c \leq a \leq b \leq d$ ($\sqsupseteq$ is defined as the converse of $\sqsubseteq$).

Definition 2.3
(Open Frames and Half-open Frames (2)). Let $\mathrm{F} = (I(T), <, >, \sqsubseteq, \sqsupseteq)$, where $I(T)$ consists either of all non-empty open intervals (a, b), or all non-empty half-open intervals $[a, b)$, and $(a, b) < (c, d)$ if $b \leq c$ (similarly for half-open intervals), and $(a, b) \sqsubseteq (c, d)$ if $c \leq a < b \leq d$ (and similarly for half-open intervals).

Theorem 2.4
Every concrete frame is a van Benthem minimal interval structure, and hence a minimal interval structure.

Proof. Straightforward. ∎

However, it is important to note that given the generality of our approach, there are minimal interval structures that are *not* concrete, in that there are intervals missing. Consider, for example, the structure $F = (\{0, 1, 2, 3\}, <, >, \sqsubseteq, \sqsupseteq)$, where $1 \sqsubseteq 0$, $3 \sqsubseteq 0$ (and their converse), but $2 \not\sqsubseteq 0$; also, $1 < 2 < 3$ (and the converse) and $1 < 3$ (and its converse, both by transitivity). Monotonicity is vacuously true and therefore F is a minimal interval structure, and indeed it is even a van Benthem minimal interval structure (vacuously); but F is obviously not concrete. We will make good use of non-concrete structures later in the article (cf. the proofs in Section 8).

We will now introduce a class of structures that will play a very important technical role throughout this article.

Definition 2.5
We say $\mathcal{F} = \langle W, <, >, \sqsubseteq, \sqsupseteq \rangle$ is a **pre-interval structure** if it satisfies the following conditons: for $<$ (and its converse, $>$) *Transitivity*; for $\sqsubseteq$ (and its converse, $\sqsupseteq$) *Reflexivity*, *Transitivity*, plus the following interaction axioms: *Right Monotonicity* and *Left Monotonicity*.

A pre-interval structure that also satisfies conjectivity is called a **van Benthem pre-interval structure**.

For pre-interval structures we no longer insist that the precedence relation is *irreflexive*, nor that the inclusion relation is *antisymmetric*.

Whilst it is obvious that every minimal interval structure is a pre-interval structure; the importance of pre-interval structures, as we will show in the subsequent sections on completeness and decidability, lies in the fact that every pre-interval structure is a bounded morphic image of some minimal interval structure. Similarly, every van Benthem pre-interval structure is a bounded morphic image of a van Benthem minimal interval structure (see Lemma 3.4). In addition, we will go on to establish that every finite pre-interval structure is a bounded morphic image of a van Benthem pre-interval structure; and that every van Benthem pre-interval structure is a bounded morphic image of a van Benthem minimal interval structure. As a consequence, we will establish that the logic of van Benthem minimal interval structures is the same as the logic of minimal interval structures.

We now introduce an interval temporal language for talking about interval structures.

Definition 2.6

The interval temporal language is defined using a countable set Φ of propositional variables denoted $p, q, r, \ldots$, plus the nullary connective: $\top$; the unary connectives: $\neg$, $\langle F\rangle$, $\langle P\rangle$, $\langle U\rangle$, $\langle D\rangle$, $[F]$, $[P]$, $[D]$, $[U]$, and the binary connectives: $\wedge, \vee$. The well-formed *formulas* of the interval temporal language are given by the rule

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \Diamond\phi \mid \Box\phi.$$

where $\Diamond \in \{\langle D\rangle, \langle U\rangle, \langle F\rangle, \langle P\rangle\}$, $\Box \in \{[D], [U], [F], [P]\}$ and $p \in \Phi$. We define $\bot = \neg\top$, and take $\rightarrow$ as abbreviation: $\psi \rightarrow \phi = \neg\psi \vee \phi$.

We now define the semantics of our interval temporal logic.

Definition 2.7

A model $\mathcal{M}$ is a structure of the form $\mathcal{M} = (W, <, >, \sqsupseteq, \sqsubseteq, V)$ where $w \in W$, $>$, $<$, $\sqsubseteq$, $\sqsupseteq$ are binary relations on W such that $<$ is the converse of $>$, and $\sqsubseteq$ is the converse of $\sqsupseteq$, and $V : \Phi \rightarrow \mathcal{P}(W)$. Then the notion of **truth** defined on $\mathcal{M}$ is defined as follows:

$\mathcal{M}, w \models p$ iff $w \in V(p)$

$\mathcal{M}, w \models \neg\phi$ iff $\mathcal{M}, w \not\models \phi$

$\mathcal{M}, w \models \phi \wedge \psi$ iff $\mathcal{M}, w \models \phi$ and $\mathcal{M}, w \models \psi$

$\mathcal{M}, w \models \phi \vee \psi$ iff $\mathcal{M}, w \models \phi$ or $\mathcal{M}, w \models \psi$

$\mathcal{M}, w \models \langle D\rangle\phi$ iff $\exists w'(w \sqsupseteq w'$ and $\mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models \langle U\rangle\phi$ iff $\exists w'(w \sqsubseteq w'$ and $\mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models \langle F\rangle\phi$ iff $\exists w'(w < w'$ and $\mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models \langle P\rangle\phi$ iff $\exists w'(w > w'$ and $\mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models [D]\phi$ iff $\forall w'(w \sqsupseteq w' \Rightarrow \mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models [U]\phi$ iff $\forall w'(w \sqsubseteq w' \Rightarrow \mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models [F]\phi$ iff $\forall w'(w < w' \Rightarrow \mathcal{M}, w' \models \phi)$

$\mathcal{M}, w \models [P]\phi$ iff $\forall w'(w > w' \Rightarrow \mathcal{M}, w' \models \phi)$

3 Completeness

In this section we present an axiom system K_{INT}, which we will show is sound and complete with respect to the class of pre-interval structures, and hence the class of minimal interval structures.

DEFINITION 3.1
The Interval Modal Logic contains besides all tautologies, and the K axiom for each modality, the following axioms:

$[trans(<)]$	$\langle F\rangle\langle F\rangle p \rightarrow \langle F\rangle p$
$[ref(\sqsubseteq)]$	$p \rightarrow \langle U\rangle p$
$[trans(\sqsubseteq)]$	$\langle U\rangle\langle U\rangle p \rightarrow \langle U\rangle p$
$[mon(<, \sqsubseteq)]$	$\langle F\rangle p \rightarrow [D]\langle F\rangle p$
$[mon(>, \sqsubseteq)]$	$\langle P\rangle p \rightarrow [D]\langle P\rangle p$
$[IA1]$	$p \rightarrow [F]\langle P\rangle p$
$[IA2]$	$p \rightarrow [P]\langle F\rangle p$
$[IA3]$	$p \rightarrow [U]\langle D\rangle p$
$[IA4]$	$p \rightarrow [D]\langle U\rangle p$

Its derivation rules are *modus ponens* and *necessitation* for $[P], [F], [D], [U]$.

THEOREM 3.2
The above axioms are valid in a frame F if and only if F is a pre-interval structure. K_{INT} is sound and complete with respect to the class of pre-interval structures.

PROOF. By appeal to Sahlqvist's Theorem [11]. ■

A K_{INT}-model is one in which the axioms of K_{INT} are valid. A K_{INT}-model is of the form $(\mathcal{F}, \mathcal{V})$ where $\mathcal{F}$ is a pre-interval structure, and V is a valuation of $\mathcal{F}$. A formula ϕ is K_{INT}-satisfiable if ϕ is satisfiable in the class of pre-interval structures.

In Section 1, we identified the class of minimal interval structures as the most general class of interval temporal structures. Hence, a pre-interval structure lacks certain desirable properties: namely, irreflexivity of the precedence relation, and antisymmetry for the inclusion relation. Unfortunately these properties are not definable in the interval temporal language; nonetheless, we can obtain the structure we want by employing a well-known technique in modal logic introduced by Segerberg [38]: *bulldozing*. The bulldozing technique utilises the concept of *bounded morphism*. The key property of bounded morphisms of models is that they preserve validity.

The notion of a bounded morphism is as follows:

DEFINITION 3.3
A **bounded morphism** [11] from a structure $(W, <, >, \sqsupseteq, \sqsubseteq)$, to another $(W', <', >', \sqsupseteq', \sqsubseteq')$, is a mapping $f: W \rightarrow W'$ such that

1. f preserves the accessibility relation $<$ (that is to say, $u < v$ implies $f(u) <' f(v)$). And similarly for the other three relations.
2. whenever $f(u) <' v'$, there is a $v \in W$ such that $f(v) = v'$ and $u < v$. And similarly for the other three relations.

A bounded morphism of Kripke models $(W, <, >, \sqsupseteq, \sqsubseteq, V)$ and $(W', <', >', \sqsupseteq', \sqsubseteq', V')$ is a bounded morphism of their underlying frames such that

3. w and $f(w)$ satisfy the same propositional atoms.

If f is a surjective-bounded morphism from $(W, <, >, \sqsupseteq, \sqsubseteq)$ to $(W', <', >', \sqsupseteq', \sqsubseteq')$, we say that $(W', <', >', \sqsupseteq', \sqsubseteq')$ is a **bounded morphic image** of $(W, <, >, \sqsupseteq, \sqsubseteq)$.

We will now establish the following: that for every (van Benthem) pre-interval structure, there exists a (van Benthem) minimal interval structure, and a surjective bounded morphism from the latter onto the former.

Lemma 3.4
Let $\mathfrak{R}$ denote the class of pre-interval structures, and $\mathfrak{I}$ denote the class of minimal interval structures. Then the following holds:

1. $\mathfrak{I} \subseteq \mathfrak{R}$,
2. For any $\mathcal{F} \in \mathfrak{R}$, there exists a $\mathcal{J} \in \mathfrak{I}$ and a surjective bounded morphism: $\mathcal{J} \to \mathcal{F}$.

Futhermore, let $\mathfrak{Q}$ denote the class of van Benthem pre-interval strutures, and $\mathfrak{K}$ denote the class of van Benthem minimal interval structures. Then the following holds:

3. $\mathfrak{K} \subseteq \mathfrak{Q}$,
4. For any $\mathcal{F} \in \mathfrak{Q}$, there exists a $\mathcal{K} \in \mathfrak{K}$ and a surjective bounded morphism: $\mathcal{K} \to \mathcal{F}$.

Proof. It is obvious that 1 holds. For proof of 2: let $\mathcal{F}^S = (S, <^S, >^S, \sqsubseteq^S, \sqsupseteq^S)$ be a pre-interval structure. We will now bulldoze $\mathcal{F}^S$ in order to obtain a minimal interval structure.

We know that $\mathcal{F}^S$ may contain $<^S$-clusters and $\sqsubseteq^S$-clusters, which we will bulldoze away. We define $\sqsubseteq^S$-clusters and $<^S$-clusters. Let $\approx_{\sqsubseteq}$ and $\approx_{<}$ be defined as follows:

$x \approx_{\sqsubseteq} y$ iff $(x \sqsubseteq^S y \wedge y \sqsubseteq^S x)$, and

$x \approx_{<} y$ iff $(x <^S y \wedge y <^S x)$

Then $\approx_{\sqsubseteq}$ is an equivalence relation on S, and $\approx_{<}$ is an equivalence relation on the set $\{w \in S : w <^S w\}$ (possibly empty). A $\sqsubseteq^S$-cluster is an equivalence class of $\approx_{\sqsubseteq}$, and a $<^S$-cluster is an equivalence class of $\approx_{<}$.

We will now impose a linear ordering on the $\sqsubseteq^S$-clusters. Let the set of $\sqsubseteq^S$-clusters be $\{C^i_{\sqsubseteq} : i \in I\}$ for some suitable I. Define a reflexive linear ordering $\preceq^i$ on each $\sqsubseteq^S$-cluster $C^i_{\sqsubseteq}$ such that each $C^i_{\sqsubseteq}$ has a maximal element with respect to $\preceq^i$.

Let $\mathcal{Q} = (U, <', >', \sqsubseteq', \sqsupseteq')$, where $U = \{(n, m) : n < m \text{ and } n, m \in \mathbb{Q}\}$, and $<' = \{((n, m), (k, l)) : n < m \leq k < l\}$, $\sqsubseteq' = \{((n, m), (k, l)) : k \leq n < m \leq l\}$. $>'$ and $\sqsupseteq'$ are defined as the converse of $<'$ and $\sqsubseteq'$ respectively. By Theorem 2.4, $\mathcal{Q}$ is a van Benthem minimal interval structure, and therefore also a minimal interval structure. In addition, let $\sqsubset' = \sqsubseteq' \wedge \neq$.

Now we will construct the product of $\mathcal{F}^S$ and $\mathcal{Q}$. Let $\mathcal{F}^S \times \mathcal{Q} = \langle S \times U, <^*, >^*, \sqsubseteq^*, \sqsupseteq^* \rangle$.

The $<^*$ relation is defined as:

$$(x, x') <^* (y, y') \iff x <^S y \wedge x' <' y'$$

and $>^*$ is defined as the converse of $<^*$.

The $\sqsubseteq^*$ relation is defined as:

$$(a, a') \sqsubseteq^* (b, b') \iff$$
$$a \sqsubseteq^S b \wedge a' \sqsubseteq' b' \wedge ((a \approx_{\sqsubseteq} b \wedge a' = b') \to \exists i (a \preceq^i b)),$$

and $\sqsupseteq^*$ is defined as the converse.

Claim 1. The product structure $\mathcal{F}^S \times \mathcal{Q}$ is a minimal interval structure.

Proof. We first check that $<^*$ is a strict ordering. Suppose $(x, x') <^* (y, y')$, then by the irreflexivity of $<'$ we have $(x, x') \neq (y, y')$. Now for transitivity suppose $(x, x') <^* (y, y') <^* (z, z')$. We want to show $(x, x') <^* (z, z')$. By the transitivity of $<^S$ we have $x <^S z$, and by the transitivity of $<'$ we have $x' <' z'$, and therefore $(x, x') <^* (z, z')$.

Now, we check that $\sqsubseteq^*$ is a partial order. The reflexivity of $\sqsubseteq^*$ follows immediately from the reflexivity of $\preceq^i$. So suppose $(x, x') \sqsubseteq^* (y, y') \sqsubseteq^* (z, z')$. We want to show that $(x, x') \sqsubseteq^* (z, z')$. If it is not the case that $(x \approx_{\sqsubseteq} z \wedge x' = z')$, then by the transitivity of both $\sqsubseteq^S$ and $\sqsubseteq'$, we have $(x, x') \sqsubseteq^* (z, z')$. If $(x \approx_{\sqsubseteq} z \wedge x' = z')$ holds, then it follows that x, y, z are in the same $\sqsubseteq$-cluster $C^i_{\sqsubseteq}$. Furthermore, we have $x' = y' = z'$. From $x \approx_{\sqsubseteq} y$ and $x' = y'$, we obtain $x \preceq^i y$, and similarly, from $y \approx_{\sqsubseteq} z$ and $y' = z'$, we get $y \preceq^i z$. By the transitivity of $\preceq^i$ we obtain $x \preceq^i z$ and therefore $(x, x') \sqsubseteq^* (z, z')$.

It remains to check that $\sqsubseteq^*$ is antisymmetric. So, suppose $(x, x') \sqsubseteq^* (y, y') \sqsubseteq^* (x, x')$. Note that by our supposition x, y are in the same $\sqsubseteq^S$-cluster $C^i_{\sqsubseteq}$. By the antisymmetry of the $\sqsubseteq'$ relation, we have $x' = y'$, and therefore $x \preceq^i y \preceq^i x$. By the antisymmetry of $\preceq^i$, we obtain $x = y$ and therefore conclude that $(x, x') = (y, y')$.

Finally, we have to check that the relations are monotonic. So, suppose $(x, x') \sqsubseteq^* (y, y') <^* (z, z')$. We want to show that $(x, x') <^* (z, z')$. Monotonicity follows from the definition.

Thus, we conclude that the product structure $\mathcal{Q} \times \mathcal{F}^S$ is a minimal interval structure. ■

Claim 2. If $\mathcal{F}^S$ is a van Benthem pre-interval structure then the product structure $\mathcal{F}^S \times \mathcal{Q}$ is a van Benthem minimal interval structure.

Proof. That $\mathcal{F}^S \times \mathcal{Q}$ is a minimal interval structure follows from Claim 1 above. It remains to show that $\mathcal{F}^S \times \mathcal{Q}$ satisfies conjectivity. Let $(x, x'), (y, y') \in S \times U$ be arbitrary, and suppose there is some (a, a') such that $(x, x') \sqsupseteq^* (a, a') \sqsubseteq^* (y, y')$. Since both $\mathcal{F}^S$ and $\mathcal{Q}$ satisfy conjectivity, we can pick (z, z') satisfying:

1. z is maximal with respect $\sqsubseteq^S$ for x and y (cf. Section 1.2),
2. z' is maximal with respect to $\sqsubseteq'$ for x' and y' (cf. Section 1.2).

Furthermore, if $w \approx_{\sqsubseteq} z$, then w is also maximal with respect to $\sqsubseteq^S$ for x and y. So we can further suppose that:

3(i) If $x \approx_{\sqsubseteq} y \approx_{\sqsubseteq} z \wedge x' = y' = z'$, then $z = min\{x, y\}$ with respect to the linear ordering $\preceq^i$,
3(ii) If $x \approx_{\sqsubseteq} z \wedge x' = z' \wedge \neg(y \approx_{\sqsubseteq} z \wedge y' = z')$, then $x = z$,
3(iii) If $y \approx_{\sqsubseteq} z \wedge y' = z' \wedge \neg(x \approx_{\sqsubseteq} z \wedge x' = z')$, then $y = z$,
3(iv) Otherwise, z is the maximal element in its $\sqsubseteq$-cluster.

First we want to establish that $(x, x') \sqsupseteq^* (z, z') \sqsubseteq^* (y, y')$. (a) If $x \approx_{\sqsubseteq} z \wedge x' = z'$, and $y \approx_{\sqsubseteq} z \wedge y' = z'$, then $x \approx_{\sqsubseteq} y$ and $x' = y'$ and so 3(i) holds. This implies that $z = min\{x, y\}$ with respect to the linear ordering $\preceq^i$ for some i, and therefore $(z, z') \in \{(x, x'), (y, y')\}$ and so we have $(x, x') \sqsupseteq^* (z, z') \sqsubseteq^* (y, y')$. (b) If $x \approx_{\sqsubseteq} z \wedge x' = z'$ and $\neg(y \approx_{\sqsubseteq} z \wedge y' = z')$, then 3(ii) holds and so $(z, z') = (x, x')$ and therefore we have $(x, x') \sqsupseteq^* (z, z') \sqsubseteq^* (y, y')$. (c) If $y \approx_{\sqsubseteq} z \wedge y' = z'$ and $\neg(x \approx_{\sqsubseteq} z \wedge x' = z')$, then 3(iii) holds and by analogous reasoning to (b) we have $(x, x') \sqsupseteq^* (z, z') \sqsubseteq^* (y, y')$. Finally (d) if neither 3(i)–3(iii) hold, then by items 1, 2 and 3(iv) it is immediate that $(x, x') \sqsupseteq^* (z, z') \sqsubseteq^* (y, y')$.

Now we want to show that for any (u, u') such that $(x, x') \sqsupseteq^* (u, u') \sqsubseteq^* (y, y')$, we have $(u, u') \sqsubseteq^* (z, z')$. So, consider any such (u, u'). By the conjectivity of both $\mathcal{F}^S$ and

$\mathcal{Q}$ we have $u \sqsubseteq^S z$ and $u' \sqsubseteq' z'$. If $u \approx_{\sqsubseteq} z \wedge u' = z'$, then by the maximality of z in its $\sqsubseteq$-cluster we have $u \preceq^i z$ for some i and so $(u, u') \sqsubseteq^* (z, z')$; otherwise, it follows immediately that $(u, u') \sqsubseteq^* (z, z')$.

Thus we conclude that the structure $\mathcal{F}^S \times \mathcal{Q}$ is a van Benthem minimal interval structure. ■

Claim 3. The mapping f: $\langle (S \times U), <^*, >^*, \sqsubseteq^*, \sqsupseteq^* \rangle \rightarrow (S, <^S, >^S, \sqsubseteq^S, \sqsupseteq^S)$, such that $f((x, x')) = x$, is a surjective bounded morphism.

Proof. Recall that a mapping f: $\langle (S \times U), <^*, >^*, \sqsubseteq^*, \sqsupseteq^* \rangle \rightarrow (S, <^S, >^S, \sqsubseteq^S, \sqsupseteq^S)$ is a *bounded morphism* if it satisfies the following conditions:

(i) f is a homomorphism with respect to the relation $<^*$ (that is, if $(x, x') <^* (y, y')$ then $f((x, x')) <^S f((y, y'))$. And similarly for the other three relations.
(ii) The *back condition* for $<^*$ (that is, if $f((x, x')) <^S y$ then there exists (y, y') such that $(x, x') <^* (y, y')$ and $f((y, y')) = y$. And similarly for the other three relations.

First, we treat the $<^*$ relation. (i) is immediate by definition of $<^*$. Proof of (ii) is straightforward: suppose $f((x, x')) <^S y$, then there is some $y' \in \mathcal{Q}$ such that $x' <' y'$, and so $(x, x') <^* (y, y')$ and $f((y, y')) = y$. The converse relation $>^*$ is treated analogously.

Now we treat the $\sqsubseteq^*$ relation. (i) is immediate by definition of $\sqsubseteq^*$. Proof of (ii) is straightforward: suppose $f((x, x')) \sqsubseteq^S y$, then take some arbitrary $y' \in \mathcal{Q}$ such that $x' \sqsubset' y'$, then $(x, x') \sqsubseteq^* (y, y')$ and $f((y, y')) = y$. The converse relation $\sqsupseteq^*$ is treated analogously.

This concludes the proof of Claim 3. ■

From Claim 1 and Claim 3, proof of item 2 of Lemma 3.4 follows immediately. From Claim 2 and Claim 3, proof of item 4 of Lemma 3.4 follows immediately. ■

Theorem 3.5
K_{INT} = Logic of pre-interval structures = Logic of minimal interval structures.

Proof. Trivially by Theorem 3.2, validity in pre-interval structures implies validity in minimal interval structures. By Lemma 3.4, validity in minimal interval structures implies validity in pre-interval structures. ■

4 Decidability

In this section, we will show that the logic of minimal interval structures is decidable. To this end, we will prove a general truth lemma stating that, for any formula ϕ, if ϕ is satisfiable in a model based on a minimal interval structure, then ϕ is satisfiable in a finite model based on a pre-interval structure. The notions introduced in this section will be used the PSPACE-completeness proof in Section 6. Furthermore, it is worth noting that the decidability proof we give here immediately establishes an EXPTIME upper bound for satisfiability problem for the logic of minimal interval structures.

Let ϕ be a formula. By pushing the negations down to the atomic level, so that negations only occur in front of atomic propositions, every formula can be rewritten equally in *negation normal form*. We will assume from now on that ϕ is in negation normal form. The benefit of considering ϕ in negation normal form is that it will enable us in Section 6 to construct a PSPACE algorithm establishing a complexity upper bound to the satisfiability problem.

Definition 4.1
Let n denote $|\phi|$ (the number of symbols in ϕ). Take Φ_ϕ to be the smallest set of well-formed formulae (*wff*) containing $\bot$ and the subformulae of ϕ and such that

1. if $[F]\beta \in \Phi_\phi$, then both $[D]\beta \in \Phi_\phi$ and $[U][F]\beta \in \Phi_\phi$,
2. if $[P]\beta \in \Phi_\phi$, then both $[D]\beta \in \Phi_\phi$ and $[U][P]\beta \in \Phi_\phi$.
3. for $\beta \neq [F]\gamma$, if $[F]\beta \in \Phi_\phi$, then $[F][F]\beta \in \Phi_\phi$.
4. for $\beta \neq [P]\gamma$, if $[P]\beta \in \Phi_\phi$, then $[P][P]\beta \in \Phi_\phi$.

Φ_ϕ is finite and $\phi \in \Phi_\phi$. It can be checked that the size of Φ_ϕ is less than or equal to *an* for some fixed a, since $\Phi_\phi \subseteq \{\psi, [D]\psi, [U][F]\psi, [U][P]\psi, [F][F]\psi, [P][P]\psi : \psi \subseteq \phi\} \cup \{\bot\} \leq 10n$.

Definition 4.2
We call $S \subseteq \Phi_\phi$ a *nice* set if it satisfies the following properties:

1. $\bot \notin S$, and if $p \in S$, then $\neg p \notin S$,
2. if $\psi \wedge \theta \in S$, then $\psi \in S$ and $\theta \in S$,
3. if $\psi \vee \theta \in S$, then $\psi \in S$ or $\theta \in S$,
4. if $[U]\beta \in S$, then $\beta \in S$,
5. if $[D]\beta \in S$, then $\beta \in S$,
6. if $[F]\beta \in S$, then $[U][F]\beta \in S$,
7. if $[P]\beta \in S$, then $[U][P]\beta \in S$,
8. if $[F]\beta \in S$ and $[F][F]\beta \in \Phi_\phi$, then $[F][F]\beta \in S$,
9. if $[P]\beta \in S$ and $[P][P]\beta \in \Phi_\phi$, then $[P][P]\beta \in S$.

Note that we do not insist on a maximality condition such as $\psi \in S \iff \neg\psi \notin S$. So, for e.g., by the above definition, $\{p\}$ is a nice set.

Definition 4.3
Given a set $\mathfrak{N}$ of nice sets, we define the relations $R^\dagger_<$, $R^\dagger_>$, $R^\dagger_\sqsubseteq$, $R^\dagger_\sqsupseteq$ as follows:

1. For any $S, S' \in \mathfrak{N}$, $SR^\dagger_< S'$ if and only if, for every *wff* $[F]\beta \in \Phi_\phi$ and every *wff* $[P]\gamma \in \Phi_\phi$, if $[F]\beta \in S$, then $[F]\beta \in S'$, $[D]\beta \in S'$ and $\beta \in S'$ *and* if $[P]\gamma \in S'$, then $[P]\gamma \in S$, $[D]\gamma \in S$ and $\gamma \in S$; *moreover* we define $R^\dagger_>$ as the converse.
2. For any $S, S' \in \mathfrak{N}$, $SR^\dagger_\sqsupseteq S'$ if and only if, for every *wff* $[D]\beta \in \Phi_\phi$ and every *wff* $[U]\gamma \in \Phi_\phi$, if $[D]\beta \in S$, then $[D]\beta \in S'$ *and* if $[U]\gamma \in S'$, then $[U]\gamma \in S$; again $R^\dagger_\sqsubseteq$ is defined as the converse.

Lemma 4.4
Let $\mathfrak{N}$ be a non-empty set of nice sets, then $(\mathfrak{N}, R^\dagger_<, R^\dagger_>, R^\dagger_\sqsubseteq, R^\dagger_\sqsupseteq)$ is a pre-interval structure.

Proof. It is straightforward to show that $R^\dagger_\sqsubseteq$ is reflexive (by item 4 & 5 in Definition 4.2), and both $R^\dagger_<$ and $R^\dagger_\sqsubseteq$ are transitive. We will show that the definitions of $R^\dagger_<$ and $R^\dagger_\sqsubseteq$ satisfy the left and right *Monotonicity* condition. For the case of right monotonicity, suppose we have that $S_0 R^\dagger_\sqsubseteq S_1$, and $S_1 R^\dagger_< S_2$. We want to show that $S_0 R^\dagger_< S_2$. Assume that $[F]\beta \in S_0$ (for $[F]\beta \in \Phi_\phi$), then $[U][F]\beta \in S_0$ (by Definition 4.2(6)). Now, since $S_0 R^\dagger_\sqsubseteq S_1$, we have that $[U][F]\beta \in S_1$. By the reflexive closure of S_1 (property 4 of nice sets), we obtain $[F]\beta \in S_1$. Since, $S_1 R^\dagger_< S_2$, we have that $[F]\beta \in S_2$, $\beta \in S_2$ and $[D]\beta \in S_2$. Now suppose $[P]\gamma \in S_2$ (for $[P]\gamma \in \Phi_\phi$). By $S_1 R^\dagger_< S_2$ we have $[P]\gamma \in S_1$, $[D]\gamma \in S_1$ and $\gamma \in S_1$. Since $S_0 R^\dagger_\sqsubseteq S_1$, we obtain $[D]\gamma \in S_0$. And by the reflexive closure of S_0 (property 5 of nice sets), we have $\gamma \in S_0$. So, it remains to show that $[P]\gamma \in S_0$. Now, since $[P]\gamma \in S_2$, we have two possible cases to consider. First, suppose $\gamma \neq [P]\alpha$ (for any α), then $[P][P]\gamma \in \Phi_\phi$ and $[P][P]\gamma \in S_2$

(by Definition 4.2(9)). Since $S_1 R^{\dagger}_{<} S_2$, we have $[P][P]\gamma \in S_1$, $[P]\gamma \in S_1$ and $[D][P]\gamma \in S_1$. Given that $S_0 R^{\dagger}_{\sqsubseteq} S_1$, we obtain $[D][P]\gamma \in S_0$ and, by the reflexive closure of S_0, we have $[P]\gamma \in S_0$, as it was required. For the second case, suppose $\gamma = [P]\alpha$, then $[P][P]\alpha \in S_2$. Since $S_1 R^{\dagger}_{<} S_2$, we have $[P][P]\alpha \in S_1$, $[P]\alpha \in S_1$ and $[D][P]\alpha \in S_1$. Given that $S_0 R^{\dagger}_{\sqsubseteq} S_1$, we obtain $[D][P]\alpha \in S_0$ and by the reflexive closure of S_0 we have $[P]\alpha \in S_0$. Since $[P][P]\alpha \in \Phi_\phi$, we have $[P][P]\alpha \in S_0$ (by Definition 4.2(9)), and therefore $[P]\gamma \in S_0$, as was required. We conclude that $S_0 R^{\dagger}_{<} S_2$. In a similar way, we can verify that left monotonicity is also satisfied. ■

Definition 4.5
Given a non-empty set $\mathfrak{N}$ of nice sets, we say $\mathfrak{N}$ is ϕ-*saturated* if it satisfies the following conditions:

1. $\exists S \in \mathfrak{N}$ with $\phi \in S$;
2. if $S \in \mathfrak{N}$ and $\diamond\psi \in S$ (for $\diamond \in \{\langle F\rangle, \langle P\rangle, \langle U\rangle, \langle D\rangle\}$), then $\exists S' \in \mathfrak{N}$ with (1) $\psi \in S'$ and (2) $S R^{\dagger}_{\theta} S'$ (for $\theta \in \{<, >, \sqsubseteq, \sqsupseteq\}$ corresponding to $\diamond$).

Lemma 4.6
Suppose ϕ is K_{INT}-satisfiable, then there is a ϕ-saturated set $\mathfrak{N}$ of nice sets.

Proof. Suppose ϕ is satisfiable in some model $\mathcal{M} = (W, R_<, R_>, R_{\sqsubseteq}, R_{\sqsupseteq}, V)$ based on a minimal interval structure. For $w \in W$, let $S_w = \{\psi : \psi \in \Phi_\phi, \mathcal{M}, w \models \psi\}$. Let $\mathfrak{N} = \{S_w : w \in \mathcal{M}\}$. A standard model-theoretic argument shows that each S_w is a nice set and $\mathfrak{N}$ is ϕ-saturated. ■

Lemma 4.7 (Truth Lemma)
If $\mathfrak{N}$ is ϕ-saturated, then there is a model $\mathcal{M} = (\mathfrak{N}, R^{\dagger}_{<}, R^{\dagger}_{>}, R^{\dagger}_{\sqsubseteq}, R^{\dagger}_{\sqsupseteq}, V)$, based on a pre-interval structure, such that for all $\psi \in \Phi_\phi$ and for all $S \in \mathfrak{N}$,

$$\psi \in S \Longrightarrow \mathcal{M}, S \models \psi$$

In particular, we have that $\mathcal{M}, S \models \phi$ for some $S \in \mathfrak{N}$. Hence, if there is a ϕ-saturated set of nice sets, then ϕ is satisfiable in a K_{INT}-model.

Proof. Suppose $\mathfrak{N}$ is ϕ-saturated. We construct a model $\mathcal{M}$ whose domain is $\mathfrak{N}$, whose relations are the relations $R^{\dagger}_{<}, R^{\dagger}_{>}, R^{\dagger}_{\sqsubseteq}, R^{\dagger}_{\sqsupseteq}$ defined on $\mathfrak{N}$ as in Definition 4.3, and whose valuation is defined as follows: let $ATOM$ be the set of propositional variables occurring in Φ_ϕ and define $V : ATOM \to \mathcal{P}(\mathfrak{N})$ by $V(p) = \{S \in \mathfrak{N} : p \in S\}$. It remains to prove the implication. This is done by induction on the structural complexity of ψ. The atomic case holds by definition of V. Now since we assume every formula in Φ_ϕ is in negated normal form, let us prove the implication for negated atoms. Suppose $\neg p \in S$. We want to show that $\mathcal{M}, S \not\models p$. Since $\neg p \in S$, we have that $p \notin S$. By the definition of V, this gives us $\mathcal{M}, S \not\models p$. The $\wedge$ and $\vee$ cases are straightforward. Now, suppose $[D]\alpha \in S$. We want to show that $\mathcal{M}, S \models [D]\alpha$. Suppose $S R^{\dagger}_{\sqsupseteq} S'$. From $[D]\alpha \in S$, we have by definition of $R^{\dagger}_{\sqsupseteq}$ that $[D]\alpha \in S'$. By the reflexive closure, we have $\alpha \in S'$. By the inductive hypothesis, we have $\mathcal{M}, S' \models \alpha$ and, as S' was arbitrary, we have $\mathcal{M}, S \models [D]\alpha$. Now suppose $\langle D\rangle\alpha \in S$. We want to show $\mathcal{M}, S \models \langle D\rangle\alpha$. By hypothesis, $\mathfrak{N}$ is a ϕ-saturated set, and so there is an S' such that $\alpha \in S'$ and $S R^{\dagger}_{\sqsupseteq} S'$. By the inductive hypothesis, we have $\mathcal{M}, S' \models \alpha$, and therefore $\mathcal{M}, S \models \langle D\rangle\alpha$. The cases of the other modalities is analogous. ■

Consequently, ϕ is K_{INT}-satisfiable if and only if there is a ϕ-saturated set.

Lemma 4.8
If ϕ is K_{INT}-satisfiable, then ϕ has a model, based on a pre-interval structure, of exponential size.

Proof. Suppose that ϕ is satisfiable. By Lemma 4.6, we have that there is a ϕ-saturated set $\mathfrak{N}$. By Lemma 4.7, $\mathcal{M} = (\mathfrak{N}, R^{\dagger}_{<}, R^{\dagger}_{>}, R^{\dagger}_{\sqsubseteq}, R^{\dagger}_{\sqsupseteq}, V)$ is a model in which ϕ is satisfied. And $\mathfrak{N} \subseteq \mathcal{P}(\Phi_{\phi})$. So $| \mathfrak{N} | \leq 2^{|\Phi_{\phi}|} \leq 2^{an}$. ■

Theorem 4.9
The logic of minimal interval structures is decidable.

Proof. In order to check whether an input formula ϕ is K_{INT}-satisfiable in a minimal interval structure, it suffices to enumerate all pre-interval structures of at most size 2^{an}. If we find a pre-interval structure in which ϕ is satisfied we output "ϕ satisfiable"; if not, then we output "ϕ unsatisfiable". The former condition is correct by the bulldozing in Lemma 3.4; and the latter is correct by Lemma 4.8. ■

5 Completeness and decidability for van Benthem minimal interval structures

The class of van Benthem pre-interval structures is obviously a subclass of the class of pre-interval structures. We will now show that for any finite pre-interval structure $\mathcal{B}$, we can construct a van Benthem pre-interval structure $\mathcal{C}$ and a mapping f from $\mathcal{C}$ onto $\mathcal{B}$ such that f is a bounded morphism. Furthermore, by Lemma 3.4, we can obtain a van Benthem minimal interval structure from any van Benthem pre-interval structure. Consequently, we establish that the logic of van Benthem minimal interval structures is decidable *and* is the same as the logic of minimal interval structures.

Proposition 5.1
If ϕ is satisfiable in a finite pre-interval structure, then there is a van Benthem pre-interval structure in which ϕ is satisfiable.

Proof. Suppose ϕ is satisfiable in a finite pre-interval structure $\mathcal{B} = (W, R_{<}, R_{>}, R_{\sqsubseteq}, R_{\sqsupseteq})$. We will construct a van Benthem pre-interval structure $\mathcal{C}$ from $\mathcal{B}$ and a mapping f from $\mathcal{C}$ onto $\mathcal{B}$ such that f is a bounded morphism. Our construction of $\mathcal{C}$ is similar in fashion to the step-by-step construction given in Venema [45].

A $\mathcal{B}$-network is of the form $\mathcal{N} = (N, <, >, \sqsubseteq, \sqsupseteq, f)$ such that $(N, <, >, \sqsubseteq, \sqsupseteq)$ is a frame, and f is a mapping from $\mathcal{N}$ onto $\mathcal{B}$. $\mathcal{N}$ is *coherent* if it satisfies

1. f is a surjective homomorphism.
2. for all $x, y \in N$, we have (1) $x < y \Longleftrightarrow f(x) R_{<} f(y)$ and (2) $x < y \Longleftrightarrow y > x$.
3. $(N, <, >, \sqsubseteq, \sqsupseteq)$ is a van Benthem pre-interval structure.

In addition, we say that $\mathcal{N}$ is *saturated* if

1. f satisfies the back condition of a bounded morphism with respect to $\sqsubseteq$ and $\sqsupseteq$. (Note that by the coherency conditions 1 & 2, f automatically satisfies the back condition with respect to $<$ and $>$).

A *perfect* $\mathcal{B}$-network is a network that is both coherent and saturated.

LEMMA 5.2
There exists a perfect $\mathcal{B}$-network.

PROOF. Let $\mathcal{N}_0 = (W, <, >, =, =, f_0)$ be the initial $\mathcal{B}$-network, where for any $x, y \in W$, we have $x < y \Longleftrightarrow f_0(x) R_< f_0(y)$ and $x < y \Longleftrightarrow y > x$; the inclusion relation is simply the equality relation between the elements of W; and f_0 is the identity function on W. Clearly $\mathcal{N}_0$ is a coherent $\mathcal{B}$-network. However, it may not be saturated. Crucially though, any witness to the imperfection of the $\mathcal{B}$-network can be repaired. Such witnesses are called *defects* and correspond to the particular violation of the saturation condition. To be more precise about what it means to repair a defect we need the notion of one network *extending* another. Let $\mathcal{N}_i = (W_i, <_i, >_i, \sqsubseteq_i, \sqsupseteq_i, f_i)$ and $\mathcal{N}_j = (W_j, <_j, >_j, \sqsubseteq_j, \sqsupseteq_j, f_j)$ be two $\mathcal{B}$-networks. We say that $\mathcal{N}_j$ *extends* $\mathcal{N}_i$, notation: $\mathcal{N}_j \rhd \mathcal{N}_i$, if W_i is a subset of W_j; the relation $<_i$ is the restriction of $<_j$ to $W_i \times W_i$ (and similarly for the other relations); and f_i is the restriction of f_j to W_i.

Claim. For any defect of a coherent $\mathcal{B}$-network $\mathcal{N}_l$ there is an $\mathcal{N}_{l+1} \rhd \mathcal{N}_l$ lacking this defect.

PROOF OF CLAIM. Let $\mathcal{N}_l = (W_l, <_l, >_l, \sqsubseteq_l, \sqsupseteq_l, f_l)$ be a coherent network and assume $\mathcal{N}_l$ has a defect with respect to $\sqsubseteq$ (the other case is analogous). Thus, we have some $w \in W_l$ such that $f_l(w) R_\sqsubseteq v'$, but there is no $v \in W_l$ such that $w \sqsubseteq_l v$ and $f_l(v) = v'$.

Take a new element v (where $v \notin W_l$), and define $\mathcal{N}_{l+1} \rhd \mathcal{N}_l$ as follows:

$$
\begin{aligned}
&W_{l+1} = W_l \cup \{v\} \\
&f_{l+1}(v) = v' \\
&x <_{l+1} y \Longleftrightarrow f_{l+1}(x) R_< f_{l+1}(y) \\
&y >_{l+1} y \Longleftrightarrow x <_{l+1} x \\
&u \sqsubseteq_{l+1} v \Longleftrightarrow u = v \vee u \sqsubseteq_l w \\
&x \sqsupseteq_{l+1} y \Longleftrightarrow y \sqsubseteq_{l+1} x.
\end{aligned}
$$

It is easy to see that f_{l+1} is surjective. We have to check that $\mathcal{N}_{l+1}$ is a coherent network. For the first coherency condition: f_{l+1} is a homomorphism with respect to the $<_{l+1}$ and $>_{l+1}$ relations by definition; in the case of $\sqsubseteq_{l+1}$, we want to show that if $x \sqsubseteq_{l+1} v$, then $f_{l+1}(x) R_\sqsubseteq f_{l+1}(v)$. Now if $x = v$, then by the definition of f_{l+1} and the reflexivity of $R_\sqsubseteq$ we have $f_{l+1}(v) R_\sqsubseteq f_{l+1}(v)$. Otherwise, we have $x \sqsubseteq_l w$. By the coherency of $\mathcal{N}_l$, we know that $f_{l+1}(x) R_\sqsubseteq f_{l+1}(w)$, and furthermore we have $f_{l+1}(w) R_\sqsubseteq f_{l+1}(v)$. By transitivity of $R_\sqsubseteq$, we have $f_{l+1}(x) R_\sqsubseteq f_{l+1}(v)$, as we had to show. The second coherency condition holds by definition. It remains to check that $\mathcal{N}_{l+1}$ is a van Benthem pre-interval structure. That $\mathcal{N}_{l+1}$ satisfies all the conditions of a pre-interval structure can be easily verified. Let us show that $\mathcal{N}_{l+1}$ satisfies conjectivity. Suppose $x, y \in \mathcal{N}_{l+1}$ such that there is some $z \in \mathcal{N}_{l+1}$ with $x \sqsupseteq_{l+1} z \sqsubseteq_{l+1} y$. If $x \sqsubseteq_{l+1} y$, then x is a *maximal* subinterval, and similarly, if $y \sqsubseteq_{l+1} x$. So, suppose $x \not\sqsubseteq_{l+1} y \not\sqsubseteq_{l+1} x$, notation $x \bot y$. We want to show that there is some *maximal* $t \in \mathcal{N}_{l+1}$ for x and y. We have a number of cases to consider. (1) If $x, y \in \mathcal{N}_l$, then we know inductively that there is a maximal $t \in \mathcal{N}_l$ such that $t \sqsubseteq_l x$ and $t \sqsubseteq_l y$. This is still true in $\mathcal{N}_{l+1}$ since $x, y \in \mathcal{N}_l$, and by the definition of $\mathcal{N}_{l+1}$ $v \not\sqsubseteq_{l+1} x$ and $v \not\sqsubseteq_{l+1} y$. (2) So, suppose we have that $x = v$ (the case where $y = v$ is identical). Then we have $y \sqsupseteq_{l+1} t \sqsubseteq_{l+1} v$. We want to show that t is still maximal with respect to v and y. By definition of the $\sqsubseteq_{l+1}$ relation we have $t \sqsubseteq_{l+1} w$. Let $u \in \mathcal{N}_{l+1}$ with $y \sqsupseteq_{l+1} u \sqsubseteq_{l+1} v$. We know that $u \neq v$ (since $v \bot y$). So, we get $u \sqsubseteq_{l+1} w$, by definition of the $\sqsubseteq_{l+1}$ on v, and $u \sqsubseteq_{l+1} y$. Now since $u \neq v$, $u \in \mathcal{N}_l$ and therefore by induction, we have that $u \sqsubseteq_{l+1} t$, as we had to show. We therefore conclude that $\mathcal{N}_{l+1}$ satisfies conjectivity and the claim is established.

Now we are ready to finish the proof of Lemma 5.2. Let $\mathcal{N}_\omega = \bigcup_{l<\omega} \mathcal{N}_l$, where unions of networks are defined in the obvious way. We need to prove that $\mathcal{N}_\omega$ is a van Benthem pre-interval structure. Again, it can be easily verified that $\mathcal{N}_\omega$ is a pre-interval structure. We will prove conjectivity. Let $x, y, z \in \mathcal{N}_\omega$ with $x \sqsupseteq_\omega z \sqsubseteq_\omega y$. Again we suppose that $x \perp y$. Take $l < \omega$ with $x, y, z \in \mathcal{N}_l$. $\mathcal{N}_l$ satisfies conjectivity. So, there is some $w \in \mathcal{N}_l$ with $x \sqsupseteq_l w \sqsubseteq_l y$ and w is maximal. We claim that this holds true for all $\mathcal{N}_{l'}$ with $l' \geq l$. The proof of the claim is by induction on l'. The base case, where $l' = l$, holds immediately. Now suppose, the claim holds for $\mathcal{N}_{l'}$, we want to show that it holds for $\mathcal{N}_{l'+1}$. In $\mathcal{N}_{l'+1}$ we still have $x \sqsupseteq_{l'+1} w \sqsubseteq_{l'+1} y$. Suppose there is some $t \in \mathcal{N}_{l'+1}$ with $x \sqsupseteq_{l'+1} t \sqsubseteq_{l'+1} y$. We require that $t \sqsubseteq_{l'+1} w$. Now there is a new element $v \in \mathcal{N}_{l'+1} \backslash \mathcal{N}_{l'}$. If $t \neq v$, then by the inductive hypothesis, we have that $t \sqsubseteq_{l'+1} w$, and the claim stands. So, suppose $t = v$, then $x \sqsupseteq_{l'+1} v \sqsubseteq_{l'+1} y$. Now v must have been introduced to either solve a $\sqsupseteq_{l'}$ or a $\sqsubseteq_{l'}$ defect. In the former case, there is some $s \in \mathcal{N}_{l'}$ such that, for all $s' \in \mathcal{N}_{l'+1}$, we have $s' \sqsupseteq_{l'+1} v \iff s' \sqsupseteq_{l'} s \vee s' = v$. Thus, by definition of $\sqsupseteq_{l'+1}$, we have $x \sqsupseteq_{l'+1} s \sqsubseteq_{l'+1} y$ and so inductively $w \sqsupseteq_{l'+1} s$. And therefore we have $v \sqsubseteq_{l'+1} w$. Thus, w is still maximal with respect to $x \perp y$ in $\mathcal{N}_{l'+1}$. In the latter case, it follows immediately that w is still maximal with respect to $x \perp y$ in $\mathcal{N}_{l'+1}$, since the way the $\sqsubseteq_{l'}$-defect is repaired (see above) ensures that $v \not\sqsubseteq_{l'+1} x$ and $v \not\sqsubseteq_{l'+1} y$. So, by induction, we conclude that w is maximal with respect to $x \perp y$ in $\mathcal{N}_\omega$. Clearly, $\mathcal{N}_\omega$ is a perfect network. ■

We will now conclude the proof of Proposition 5.1. It is clear that $\mathcal{B}$ is a bounded morphic image of $\mathcal{N}_\omega$. Let V be a valuation on $\mathcal{B}$, we define a valuation V^+ on $\mathcal{N}_\omega$ as follows: for propositional variable p, $V^+(p) = f_\omega^{-1}(V(p))$. Then Proposition 5.1 follows immediately from Lemma 5.2 and also from f_ω being an onto bounded morphism. ■

Now, by appealing to Lemma 3.4, we have that, if ϕ is satisfiable in a van Benthem pre-interval structure, then ϕ is satisfiable in a van Benthem minimal interval structure. Tying everything together, we get that the logic of minimal interval structures is the *same* as the logic of van Benthem minimal interval structures.

Theorem 5.3
K_{INT} = Logic of minimal interval structures = Logic of van Benthem minimal interval structures.

Proof. By Theorem 3.2, Lemma 4.8, Proposition 5.1 and Lemma 3.4. ■

Theorem 5.4
The logic of van Benthem minimal interval structures is decidable.

Proof. By Theorem 4.9 and Theorem 5.3. ■

6 Complexity

In this section, we will address complexity issues for the satisfiability problem for K_{INT}. We will show that the problem is PSPACE-complete.

In [28], Ladner showed that every modal logic between K and $S4$ is PSPACE-hard. Subsequently, Spaan [41] constructed polynomial space bounded algorithms to show that the temporal logics $K4_t$ and $S4_t$ are PSPACE-complete.

We are interested in the complexity class of the following set.

Definition 6.1
$\text{SAT}(K_{\text{INT}}) = \{\phi : \phi \text{ is a satisfiable } K_{\text{INT}} \text{ formula}\}$.

A straightforward reduction of the satisfiability for *S*4 establishes the PSPACE-hardness of SAT(K_{INT}).

Theorem 6.2
The satisfiability problem for K_{INT} is PSPACE-hard.

Proof. Observe that any structure $F = (W, <, >, \sqsubseteq, \sqsupseteq)$ with $\sqsubseteq = \sqsupseteq^{-1}$ transitive and reflexive, and $<=>= \emptyset$, is a pre-interval structure. It follows that K_{INT} is a conservative extension of *S*4. Hence it is PSPACE-hard. ■

It remains to prove a PSPACE upper bound. Due to the following relationship it is sufficient to present a non-deterministic PSPACE algorithm in order to prove that a problem is in deterministic PSPACE.

Fact 6.3 (Savitch's Theorem [30])
PSPACE=NPSPACE.

We could adapt the strategy employed by Spaan [41], which is itself an extension of Ladner's approach [28], in order to establish a polynomial space bound for K_{INT}. Instead, we will follow a different approach (exploiting similar ideas), presented in [25], which gives a tableau algorithm establishing a polynomial space bound for the description logic $\mathcal{ALCNI}_{R^+}$, a description logic with transitive and inverse roles.

We will now try to explain the intuition behind this approach. A tableau algorithm checks for the satisfiability of a formula ϕ by trying to construct a model of ϕ. The model is represented by a tree in which each node x is labelled with a set of formulas $L(x)$, and each edge is labelled with a relation.

The algorithm starts with a single node labelled by $\{\phi\}$, and proceeds by repeatedly applying a set of *expansion rules* that recursively decompose the formulas in the node labels; new edges and nodes are added as required in order to satisfy $\Diamond\psi$ formulas. The construction terminates either when none of the rules can be applied in a way that extends the tree, or when the discovery of obvious contradictions demonstrates that ϕ has no model.

However, termination is not guaranteed for logics that include transitive relations, as the expansion rules can introduce new formulas that have the same size as the decomposed formula. In particular, $[\xi]\psi$ formulas, where ξ is a transitive relation, are dealt with by propagating the whole formula across ξ-labelled edges.

This problem is dealt with by *blocking*: halting the expansion process when a cycle is detected [6]. For logics without inverse relations, the general procedure is to check the label of each new node y, and if it is a *subset* of the label of an existing node x, then no further expansion of y is performed: x is said to block y.

Blocking is, however, more problematic when inverse relations are added to the logic. Because relations are now bi-directional, it is no longer possible to establish a block on a once and for all basis when a new node is added to the tree. This is because further expansion in other parts of the tree could lead to the labels of the blocking and/or blocked nodes being extended and the block being invalidated. The problem posed by inverse relations is resolved by introducing the concept of *dynamic blocking*.

Dynamic blocking allows blocks to be dynamically established and broken as the expansion progresses, whilst continuing to expand $[\xi]\psi$ formulas in the labels of the blocked nodes. Furthermore, in addition to the label L, each node now has a second label B, where the latter is always a subset of the former. The label L contains complete information, whereas B only contains information relevant to blocking. Examples of how dynamic blocking works can be found in [25]. Alongside the blocking technique, they also use the notion of the maximal *length* of a formula to show that every time an alternation in the relation occurs, the maximal length of the formula decreases. Bringing both together, they are able to show that blocking occurs if the maximum length of a branch in the completion tree, without any alternating relations, exceeds a polynomial bound. Otherwise, when the relation alternates, the length of the formula decreases, which can only happen a linear number of times in the size of the original formula. Thus, a polynomial space bound for the description logic $\mathcal{ALCNI}_{R^+}$ is established.

Our reason for choosing the tableau method is that the procedure is potentially very efficient and susceptible to practical testing. Because the tableau algorithm does not use maximal consistent sets, it has real implementation benefits in terms of practical testing. Also the notion of dynamic blocking, which allows nodes to be blocked and then unblocked, as the information stored changes, provides a sophisticated guiding technique in the search for a solution.

However, as it stands, the tableau algorithm for $\mathcal{ALCNI}_{R^+}$ is inadequate for our purposes. This is due to the presence of interacting relations in our logic. Whereas, with the $\mathcal{ALCNI}_{R^+}$-algorithm, any alternation in the relation led to a decrease in the length of the formula; because of the monotonicity condition, this is no longer a guarantee for our logic. In fact, due to the monotonicity condition, expansion of the label set will lead to an *increase* in the length of the formula.

We resolve this problem by extending the algorithm. This will involve the following: first, we will store more information in the label set L than simply propositional consequences, in order to ensure that the relations satisfy the necessary properties, such as monotonicity. Secondly, the notion of the maximal length of a formula is of no use to us; since we store more information into the L set, the expansion rules may actually *increase* the length of a formula. So instead, we use the notion of the maximal *rank* of a formula. Intuitively, the rank of a formula, unlike its length, is unaffected by certain expansion rules. This allows us to show that certain patterns of alternating relations do indeed lead to a decrease in the rank of a formula; however, because of the monotonicity relation, even the rank of a formula need not always decrease when we have a change in the relation. Fortunately, a closer examination allows us to establish that, even in such cases, a polynomial bound on the length of a path in the completion tree can be established. Though the presence of interacting relations does, predictably, lead to a substantial increase in the polynomial bound for the K_{INT} algorithm, in comparison to the $\mathcal{ALCNI}_{R^+}$ algorithm.

Having reassured ourselves that the reader now has some intuition concerning our intentions, we will with undue haste, proceed to the details.

6.1 The K_{INT} algorithm

Like other tableaux algorithms, the K_{INT} algorithm tries to prove the satisfiability of a formula ϕ by constructing a model of ϕ. The model is represented by a *label tree*. This is a tree $\mathbf{T} = (T, L, B, E, \mathcal{L})$ in which each node $x \in T$ is labelled with two sets $L(x)$ and $B(x)$, where both sets are subsets of Φ_ϕ, and $E \subseteq T \times T$ such that (T, E) is a directed rooted tree. Furthermore, each pair (x, y) such that $(x, y) \in E$ or $(y, x) \in E$ is labelled $\mathcal{L}((x, y)) = \xi$ for $\xi \in \{F, P, U, D\}$, and $Inv(\mathcal{L}((x, y))) = \mathcal{L}((y, x))$ where $Inv(F) = P$ and $Inv(U) = D$ and the converse. Edges are added when expanding $\Diamond\psi$ formulas; they correspond to relationships between pairs of individuals and are always directed from the root node to the leaf nodes. The algorithm expands the tree by extending $L(x)$ (and possibly $B(x)$) for some node x or by adding new leaf nodes.

If nodes x and y are connected by an edge $E(x, y)$, then y is called a *successor* of x and x is called a *predecessor* of y. If $\mathcal{L}((x, y)) = \xi$, then y is called an ξ-*successor* of x and x is called an $Inv(\xi)$-*predecessor* of y. *Ancestor* is the transitive closure of *predecessor* and *descendant* is the transitive closure of *successor*. A node y is called an ξ-*neighbour* of a node x if either y is an ξ-*successor* of x or y is an ξ-*predecessor* of x.

A node y is *blocked* if there exists $x \in \{z : z = y \vee z \text{ is an ancestor of } y\}$ such that

$$B(y) \subseteq L(x) \text{ and } L(y)/Inv(\xi) = L(x)/Inv(\xi)$$

where if z is a predecessor of y then $\mathcal{L}((z, y)) = \xi$. The set $L(y)/Inv(\xi)$ is defined by

$$L(y)/Inv(\xi) = \{[Inv(\xi)]\psi : [Inv(\xi)]\psi \in L(y)\}.$$

A *label tree* constructed by the algorithm starting with ϕ at its root is called a *completion tree*. A completion tree **T** is said to contain a *clash* if, for a node x in T, it holds that $\perp \in L(x)$ or $\{\psi, \neg\psi\} \subseteq L(x)$. The completion tree is *complete* when for some node x, $L(x)$ contains a clash or when none of the rules are applicable. Given that the algorithm is non-deterministic, we say that the algorithm accepts ϕ if and only if there exists a complete, clash-free, completion tree for ϕ.

The algorithm initialises a tree **T** to contain a single node x_0, the *root* node, with $L(x_0) = B(x_0) = \{\phi\}$, where ϕ, in negation normal form, is the formula to be tested for satisfiability. **T** is then expanded by repeatedly applying the rules from Table 1.

For the sake of brevity, we shall use $\Box$ to range over the set $\{[F], [P], [D], [U]\}$, and henceforth refer to the following rules collectively as *local* $\Box$-*rules*: $Ref_{[D]}$, $Ref_{[U]}$, $Mon_{[U]}$, $Trans_{[F]}$, $Trans_{[P]}$. Notice that the application of the *local* $\Box$-*rules*, the $\vee$- and $\wedge$-rules only affect the label set L. Similarly, the $\Box_+$-rules only store information into the B set, when these rules are applied "downwards". When the expansion rules are applied, the label set L (and the B set) are updated accordingly – this is the procedural semantics of '$\rightarrow$' in the rules in Table 1.

6.2 Soundness and completeness

The soundness and completeness of the algorithm will be demonstrated by proving that, for a formula ϕ, it always terminates and that it accepts ϕ if and only if ϕ is K_{INT}-satisfiable.

LEMMA 6.4
Let **T** be a completion tree obtained by applying the expansion rules to a formula ϕ, then, for every node x in **T**, $B(x) \subseteq L(x)$.

PROOF. By simple induction on the number of rule applications. ■

LEMMA 6.5
For any formula ϕ, the tableau algorithm terminates.

PROOF. Let $m = |\Phi_\phi|$. We know m is linear in the length of ϕ. Termination is a consequence of the following properties of the expansion rules:

1. The expansion rules never remove nodes from the tree or formulas from node labels.
2. Successors are only generated for formulas of the form $\Diamond\psi$ (for $\Diamond \in \{\langle F\rangle, \langle P\rangle, \langle U\rangle, \langle D\rangle\}$), and for any node each of these formulas triggers the generation of at most

TABLE 1. Tableaux expansion rules

$\wedge$-*rule*:	if 1. $\psi \wedge \theta \in L(x)$ and 2. $\{\psi, \theta\} \not\subseteq L(x)$ then $L(x) \to L(x) \cup \{\psi, \theta\}$
$\vee$-*rule*:	if 1. $\psi \vee \theta \in L(x)$ and 2. $\{\psi, \theta\} \cap L(x) = \emptyset$ then $L(x) \to L(x) \cup \{\gamma\}$ for some $\gamma \in \{\psi, \theta\}$
$Ref_{[\xi]}$-*rule* (for $\xi \in \{D, U\}$):	if 1. $[\xi]\psi \in L(x)$ and 2. $\psi \notin L(x)$ then $L(x) \to L(x) \cup \{\psi\}$
$Mon_{[U]}$-*rule*:	if 1. $[\xi]\psi \in L(x)$(for $\xi \in \{F, P\}$) and 2. $[U][\xi]\psi \notin L(x)$ then $L(x) \to L(x) \cup \{[U][\xi]\psi\}$
$Trans_{[\xi]}$-*rule* (for $\xi \in \{F, P\}$):	if 1. $[\xi]\psi \in L(x)$ and 2. $\psi \neq [\xi]\beta$ and 3. $[\xi][\xi]\psi \notin L(x)$ then $L(x) \to L(x) \cup \{[\xi][\xi]\psi\}$
$[\xi]_+$-*rule* ($\xi \in \{F, P\}$):	if 1. $[\xi]\psi \in L(x)$ and either 2. there is a ξ-successor y of x with $\{[\xi]\psi, \psi, [D]\psi\} \not\subseteq B(y)$ then $L(y) \to L(y) \cup \{[\xi]\psi, \psi, [D]\psi\}$ and $B(y) \to B(y) \cup \{[\xi]\psi, \psi, [D]\psi\}$, or 2′. there is a ξ-predecessor y of x with $\{[\xi]\psi, \psi, [D]\psi\} \not\subseteq L(y)$ then $L(y) \to L(y) \cup \{[\xi]\psi, \psi, [D]\psi\}$
$[\xi]_+$-*rule* (for $\xi \in \{D, U\}$):	if 1. $[\xi]\psi \in L(x)$ and either 2. there is a ξ-successor y of x with $[\xi]\psi \notin B(y)$ then $L(y) \to L(y) \cup \{[\xi]\psi\}$ & $B(y) \to B(y) \cup \{[\xi]\psi\}$ or 2′. there is a ξ-predecessor y of x with $[\xi]\psi \notin L(y)$ then $L(y) \to L(y) \cup \{[\xi]\psi\}$
$\langle\xi\rangle_+$-*rule* (for $\xi \in \{F, P, D, U\}$):	if 1. $\langle\xi\rangle\psi \in L(x)$, x is not blocked and no other rule is applicable to any of its ancestors, and 2. x has no ξ-neighbour y with $\psi \in B(y)$ then create a new node y with $\mathcal{L}((x, y)) = \xi$ and $L(y) = B(y) = \{\psi\}$

one successor. Since Φ_ϕ contains at most $m \Diamond \psi$ formulas, the out-degree of the tree is bounded by m.

3. Nodes are labelled with nonempty subsets of Φ_ϕ. If a path π is of length at least 2^{m+1}, then there are two nodes x and y on π, with $L(x) = L(y)$ and $B(x) = B(y)$, and blocking occurs. Since a path on which nodes are blocked cannot become longer, paths are of length at most 2^{m+1}.

■

The following lemma implies soundness of the tableau algorithm.

Lemma 6.6
If the expansion rules can be applied to a formula ϕ such that they yield a complete and clash-free completion tree, then ϕ has a K_{INT}-model.

Proof. Let **T** be the complete and clash-free completion tree constructed by the tableau algorithm for ϕ. Define:

$$S = \{x \mid x \text{ is a node of T, and } x \text{ is not blocked}\},$$
$$\mathfrak{N} = \{L(x) \mid x \in S\}, \text{ where } L \text{ is the labelling in } \mathbf{T}.$$

We have to show that each $L(x)$ is a nice set (cf. Definition 4.2) and that $\mathfrak{N}$ is a ϕ-saturated set (cf. Definition 4.5). Item 1 of Definition 4.2 is satisfied since T is clash-free. Item 2 is satisfied by the $\wedge$-rule. Item 3 is satisfied by the $\vee$-rule. Item 4 is satisfied by the $Ref_{[U]}$-rule. Item 5 is satisfied by the $Ref_{[D]}$-rule. Items 6 and 7 are satisfied by the $Mon_{[U]}$-rule. Item 8 is satisfied by the $Trans_{[F]}$-rule. Finally, item 9 is satisfied by the $Trans_{[P]}$-rule. It remains to show that $\mathfrak{N}$ is a ϕ-saturated set. First, we show that $\phi \in L(x)$ for some $x \in S$. We have that $\phi \in L(x_0)$ for the root x_0 of **T** and, as x_0 has no predecessors, it cannot be blocked. Hence $\phi \in L(x)$ for some $x \in S$. Now we want to show that for any $x \in S$ with $\Diamond \psi \in L(x)$, we have some $y \in S$ with (1) $\psi \in L(y)$ and (2) $L(x)R^{\dagger}_{\xi}L(y)$ (for $\xi \in \{<, >, \sqsubseteq, \sqsupseteq\}$ corresponding to $\Diamond$).

First, suppose $\langle F\rangle\psi \in L(x)$. The case where $\langle P\rangle\psi \in L(x)$ is analogous. Then, seeing that x is not blocked, the $\langle F\rangle_+$-rule ensures that some neighbour y of x satisfies $\psi \in B(y)$. There are then two cases to consider, either:

(a) a $<$-predecessor y with $\psi \in B(y) \subseteq L(y)$ (by Lemma 6.4). Because y is a $<$-predecessor of x it cannot be blocked, so $y \in S$. Now suppose $[F]\theta \in L(x)$. Then the $[F]_+$-rule will ensure that $\{[F]\theta, \theta, [D]\theta\} \subseteq L(y)$. Now suppose $[P]\gamma \in L(y)$. Since y is a $<$-predecessor of x, we have that x is an $>$-successor of y, and by the $[P]_+$-rule, we have $\{[P]\gamma, \gamma, [D]\gamma\} \subseteq L(x)$. Therefore $L(x)R^{\dagger}_{<}L(y)$.

(b) a $<$-successor y with $\psi \in B(y) \subseteq L(y)$ (again, by Lemma 6.4). If y is not blocked, then by analogous reasoning to case (a) we have $y \in S$ and $L(x)R^{\dagger}_{<}L(y)$. Otherwise, y is blocked by some z which is an ancestor of y with $B(y) \subseteq L(z)$ and $\psi \in L(z)$. Now, we have that either z is equal to x or z is also an ancestor of x. Moreover, z is not blocked and z is in S. So suppose $[F]\theta \in L(x)$. As y is a $<$-successor of x, the $[F]_+$-rule ensures that $\{[F]\theta, \theta, [D]\theta\} \subseteq B(y)$, and therefore $\{[F]\theta, \theta, [D]\theta\} \subseteq L(z)$. Now suppose $[P]\gamma \in L(z)$. By the definition of blocking, we have $L(z)\backslash Inv(<) = L(y)\backslash Inv(<)$, and therefore $[P]\gamma \in L(y)$. As y is a $<$-successor of x, x is a $>$-predecessor of y, and the $[P]_+$-rule ensures that $\{[P]\gamma, \gamma, [D]\gamma\} \subseteq L(x)$. Thus $L(x)R^{\dagger}_{<}L(z)$.

Now, suppose $\langle D\rangle\psi \in L(x)$. The case where $\langle U\rangle\psi \in L(x)$ is analogous. Then, seeing that x is not blocked, the $\langle D\rangle_+$-rule ensures that some neighbour y of x satisfies $\psi \in B(y)$. There are then two cases to consider, either:

(c) a $\sqsupset$-predecessor y with $\psi \in B(y) \subseteq L(y)$ (by Lemma 6.4). Because y is a $\sqsupset$-predecessor of x it cannot be blocked, so $y \in S$. Now suppose $[D]\theta \in L(x)$. Then the $[D]_+$-rule will ensure that $[D]\theta \in L(y)$. Now suppose $[U]\gamma \in L(y)$. Since y is a $\sqsupset$-predecessor of x, we have that x is an $\sqsubseteq$-successor of y, and by the $[U]_+$-rule, we have $[U]\gamma \in L(x)$. Therefore $L(x)R^{\dagger}_{\sqsupseteq}L(y)$.

(d) a $\sqsupset$-successor y with $\psi \in B(y) \subseteq L(y)$ (again, by Lemma 6.4). If y is not blocked, then by analogous reasoning to case (c) we have $y \in S$ and $L(x)R^{\dagger}_{\sqsupseteq}L(y)$. Otherwise, y is blocked by some z which is an ancestor of y with $B(y) \subseteq L(z)$ and $\psi \in L(z)$. Now, we have that either z is equal to x or z is also an ancestor of x. Moreover, z is not blocked and z is in S. So suppose $[D]\theta \in L(x)$. As y is a $\sqsupset$-successor of x, we have by the $[D]_+$-rule that $[D]\theta \in B(y)$, and therefore $[D]\theta \in L(z)$. Now suppose $[U]\gamma \in L(z)$. By the definition of blocking, we have $L(z)\backslash Inv(\sqsupseteq) = L(y)\backslash Inv(\sqsupseteq)$, and therefore $[U]\gamma \in L(y)$. As y is a $\sqsupset$-successor of x, x is a $\sqsubseteq$-predecessor of y, and the $[U]_+$-rule ensures that $[U]\gamma \in L(x)$. Thus $L(x)R^{\dagger}_{\sqsupseteq}L(z)$.

Finally, by appealing to Lemma 4.7 we know that there is a K_{INT}-model for ϕ. ■

Lemma 6.7
If ϕ has a K_{INT}-model, then there exists a complete and clash-free completion tree for ϕ.

Proof. The proof is a straightforward adaptation of the proof given in [25]. Let $T = (S, R^{\dagger}_{<}, R^{\dagger}_{>}, R^{\dagger}_{\sqsubseteq}, R^{\dagger}_{\sqsupseteq}, V)$ be a K_{INT}-model for ϕ. Using T, we trigger the application of the expansion rules such that they yield a completion tree **T** that is both complete and clash-free. We start with **T** consisting of a single node x_0, the root, with $B(x_0) = L(x_0) = \{\phi\}$.

T is a model, hence there is some $s_0 \in S$ with $T, s_0 \models \phi$. When applying the expansion rules to **T**, the application of the non-deterministic $\vee$-rule is driven by the labelling in the model T. To this purpose, we define a mapping f which maps the nodes of **T** to elements of S, and we steer the application of the $\vee$-rule such that $L(x) \subseteq V(f(x))$ holds for all nodes x of the completion tree.

More precisely, we define f inductively as follows:

- $f(x_0) = s_0$.
- If $f(x_i) = s_i$ is already defined, and a successor y of x_i was generated for $\Diamond\psi \in L(x_i)$, then $f(y) = t$ for some $t \in S$ with $\psi \in V(t)$ and $(s_i, t) \in R^{\dagger}_{\xi}$ (for $\xi \in \{<, >, \sqsubseteq, \sqsupseteq\}$ corresponding to $\Diamond$).

To make sure that we have $L(x_i) \subseteq V(f(x_i))$, we use the modified $\vee'$-rule given below instead of the $\vee$-rule. The expansion rules given in Table 1 with the $\vee$-rule replaced by the $\vee'$-rule are called the *modified* expansion rules in the following.

$\vee'$-rule: If 1. $\psi \vee \theta \in L(x)$, x is not blocked, and
2. $\{\psi, \theta\} \cap L(x) = \emptyset$
then $L(x) \rightarrow L(x) \cup \{\delta\}$ for some $\delta \in \{\psi, \theta\} \cap V(f(x))$

Whereas the original $\vee$-rule presented a choice between disjuncts, the $\vee'$-rule picks a particular disjunct, and is therefore consistent with the original $\vee$-rule, since for a tree **T** generated using the modified expansion rules, the original expansion rules can be applied in such a way that they also yield **T**. Hence Lemma 6.5 and Lemma 6.6 still apply, and thus using the $\vee'$-rule instead of the $\vee$-rule preserves termination and soundness.

We will now show by induction that, if $L(x) \subseteq V(f(x))$ holds for all nodes x in **T**, then the application of an expansion rule preserves this subset-relation. To start with, we clearly have $\{\phi\} = L(x_0) \subseteq L(s_0)$.

If the $\wedge$-rule can be applied to x in **T** with $\gamma = \psi \wedge \theta \in L(x)$, then ψ, θ are added to $L(x)$. Since T is a model, $\{\psi, \theta\} \subseteq V(f(x))$, and hence the $\wedge$-rule preserves $L(x) \subseteq V(f(x))$.

If the $\vee'$-rule can be applied to x in **T** with $\gamma = \psi \vee \theta \in L(x)$, then there exists $\delta \in \{\psi, \theta\}$ such that δ is in $V(f(x))$, and δ is added to $L(x)$ by the $\vee'$-rule. Hence the $\vee'$-rule preserves $L(x) \subseteq V(f(x))$.

If the $Ref_{[D]}$-rule can be applied to x in **T** with $[D]\psi \in L(x)$, then $\psi \in V(f(x))$ since T is a model, and ψ is added to $L(x)$ by the $Ref_{[D]}$-rule. Similarly for the $Ref_{[U]}$-rule. Hence both the $Ref_{[D]}$-rule and the $Ref_{[U]}$-rule preserve $L(x) \subseteq V(f(x))$.

If the $Mon_{[U]}$-rule can be applied to x in **T** with (say) $[F]\psi \in L(x)$, then $[U][F]\psi \in V(f(x))$ since T is a model and $[U][F]\psi$ is added to $L(x)$ by the $Mon_{[U]}$-rule. Similarly for $[P]\psi$. Hence the $Mon_{[U]}$-rule preserves $L(x) \subseteq V(f(x))$.

If the $Trans_{[F]}$-rule can be applied to x in **T** with $[F]\psi \in L(x)$, then $[F][F]\psi \in V(f(x))$ since T is a model, and $[F][F]\psi$ is added to $L(x)$ by the $Trans_{[F]}$-rule. Similarly for the $Trans_{[P]}$-rule. Hence both the $Trans_{[F]}$-rule and the $Trans_{[P]}$-rule preserve $L(x) \subseteq V(f(x))$.

If the $[F]_+$-rule can be applied to x in **T** with $[F]\psi \in L(x)$ and with y a $<$-neighbour of x, then $(f(x), f(y)) \in R^{\dagger}_{<}$, and thus $\{[F]\psi, \psi, [D]\psi\} \subseteq L(f(y))$. The $[F]_+$-rule adds $\{[F]\psi, \psi, [D]\psi\}$ to $L(y)$ and thus preserves $L(y) \subseteq L(f(y))$. The case where the $[P]_+$-rule is applicable is similar.

If the $[D]_+$-rule can be applied to x in **T** with $[D]\psi \in L(x)$ and with y a $\sqsupseteq$-neighbour of x, then $(f(x), f(y)) \in R^{\dagger}_{\sqsupseteq}$, and thus $[D]\psi \in L(f(y))$. The $[D]_+$-rule adds $[D]\psi$ to $L(y)$ and thus preserves $L(y) \subseteq L(f(y))$. The case where the $[U]_+$-rule is applicable is similar.

If any of the $\diamond_+$ rules can be applied to x in **T** with $\diamond\psi \in L(x)$, then $\diamond\psi \in V(f(x))$ and there is some $t \in S$ with $(f(x), t) \in R^{\dagger}_{\xi}$ and $\psi \in V(t)$. The $\diamond_+$-rules create a new ξ-successor y of x, with $\mathcal{L}((x, y)) = \xi$, for which $f(y) = t$ for some t with $\psi \in V(t)$. Hence we have $L(y) = \{\psi\} \subseteq L(f(y))$.

Summing up, the tableau-construction triggered by T terminates with a complete tree, and since $L(x) \subseteq V(f(x))$ holds for all nodes x in **T**, **T** is clash-free. ■

6.3 Complexity of the K_{INT} algorithm

We will now turn our attention to the complexity of the tableaux algorithm in terms of memory consumption.

Take a clash-free completion tree. We will from now only consider the nodes of this tree. In the subsequent lemmas, we establish a polynomial bound on the length of paths in the completion tree. It then only remains to show that such a tree can be constructed only using polynomial space.

We will start by mentioning some basic facts which follow immediately by inspection of the tableau rules.

For each node x of the completion tree, $B(x)$ only contains two kinds of formulas: the formula which triggered the generation of the node x, denoted by C_x, and formulas propagated *down* the completion tree by the $\Box_+$-rules. Also, by Lemma 6.4, $B(x) \subseteq L(x)$ holds for any node in the completion tree.

Let us fix some notation for future reference. Let x and y be successive nodes of the completion tree such that x precedes y. Let C_x denote the formula that caused the generation of the node x. Define

$$A(y) = \begin{cases} \{[\xi]\theta : [\xi]\theta \in L(x)\}, & \text{if } \xi \in \{U, D\} \\ \{\theta, [\xi]\theta, [D]\theta : [\xi]\theta \in L(x)\}, & \text{if } \xi \in \{F, P\} \end{cases}$$

Hence, the set $A(y)$ contains only formulas inserted by application of the $\Box_+$-rules. So, we have $B(y) \subseteq A(y) \cup \{C_y\}$. Furthermore, we define the *rank* of a formula ϕ ($r(\phi)$) inductively on the complexity of ϕ as follows:

$r(\top) = 0$,
$r(p) = 0$, for all propositional variables,
$r(\neg\psi) = r(\psi)$,
$r(\psi \wedge \theta) = r(\psi \vee \theta) = \max(r(\psi), r(\theta))$,
$r(\Diamond\psi) = 1 + r(\psi)$,
$r(\Box\psi) = 1 + r(\psi)$ if $\psi \neq [\xi]\theta$, for $\xi \in \{F, P\}$

$$r(\Box[\xi]\theta)(\text{for } \xi \in \{F, P\}) = \begin{cases} 1 + r([\xi]\theta) & \text{if } \Box = [Inv(\xi)] \\ r([\xi]\theta) & \text{otherwise} \end{cases}$$

For a node x, we define $r(x) = \max\{r(\psi) : \psi \in L(x)\}$.

Definition 6.8
Let $n \geq 1$ and $x_0, x_1, \ldots, x_n \in T$ be such that x_{i+1} is a successor of x_i for each $i < n$. Then we write $[x_0, x_n)$ for the set $\{x_0, \ldots, x_{n-1}\}$ and refer to such sets as *right-open* intervals.

The above notation allows us to represent a succession of nodes as the disjoint union of right-open intervals, for example $[x_0, x_6) = [x_0, x_2) \cup [x_2, x_6)$. This will prove useful in examining the length of paths in the completion tree.

Lemma 6.9
Let x_0 and x_1 be successive nodes such that $\mathcal{L}((x_0, x_1)) = \xi$ for $\xi \in \{U, D, F, P\}$. Then $r(x_0) \geq r(x_1)$.

Proof. Let ψ be a maximal rank formula in $L(x_1)$. It is easy to see that the application of the local $\Box$-rules does not increase the rank of a formula. To get a flavour of this, suppose that ψ was inserted into $L(x_1)$ by a local $\Box$-rule, say the $Trans_{[F]}$-rule. Then $\psi = [F]\beta$ and $\beta \neq [F]\alpha$. By the application of the $Trans_{[F]}$-rule, we get $[F][F]\beta \in L(x_1)$. By the definition of rank, we have that $r([F]\psi) = r(\psi)$. By analogous reasoning we can show that the application of the remaining local $\Box$-rules also does not increase the rank of a formula. So we can assume that ψ was inserted into $L(x_1)$ by an application of the $\Diamond_+$-rules or the $\Box_+$-rules. In the former case, we have $\Diamond\psi \in L(x_0)$ and therefore $r(x_0) > r(x_1)$. In the latter case, we have two possibilities to consider: either (1) $\xi \in \{U, D\}$, or (2) $\xi \in \{F, P\}$. Suppose (1) holds. Then $\psi = [\xi]\theta$ and $[\xi]\theta \in L(x_0)$, and therefore $r(x_0) \geq r(x_1)$. For (2), suppose $\xi = F$ (the case where $\xi = P$ is similar), we then have $\psi \in \{[F]\theta, \theta, [D]\theta\}$ and $[F]\theta \in L(x_0)$. We will show that $r([F]\theta) \geq r([D]\theta)$ and $r([F]\theta) \geq r(\theta)$, and therefore $r(x_0) \geq r(x_1)$. Suppose $\theta \neq [F]\rho$ and

$\theta \neq [P]\rho$, then $r([D]\theta) = 1 + r(\theta) = r([F]\theta)$. If $\theta = [F]\rho$, then $r([D]\theta) = r([F]\theta) = r(\theta)$. Finally, if $\theta = [P]\rho$, then $r([F]\theta) > r([D]\theta) = r(\theta)$. ■

Lemma 6.10
Let $[x_0, x_2)$ be a right-open interval such that $\mathcal{L}((x_0, x_1)) = U$ and $\mathcal{L}((x_1, x_2)) = D$. Then $r(x_0) > r(x_2)$.

Proof. By Lemma 6.9 we have $r(x_0) \geq r(x_1) \geq r(x_2)$. Let ψ be a maximal rank formula in $L(x_2)$. Since the application of the local $\Box$-rules does not increase the rank of a formula, we can assume without loss of generality that ψ was inserted into $L(x_2)$ by an application of either the $\langle D\rangle_+$-rule or the $[D]_+$-rule to $L(x_1)$. In the case of the former, we have $\langle D\rangle\psi \in L(x_1)$, and therefore $r(x_1) > r(x_2)$, and so $r(x_0) > r(x_2)$. In the latter case, we have that $\psi = [D]\theta$ and $\psi \in L(x_1)$. Now if ψ is not a maximal rank formula in $L(x_1)$ then we have $r(x_1) > r(x_2)$, and so $r(x_0) > r(x_2)$. So, suppose $\psi = [D]\theta$ is a maximal rank formula in $L(x_1)$. By supposition, we now have that ψ could not have been inserted into $L(x_1)$ by an application of the $[U]_+$-rule. The only possibility is that ψ was inserted into $L(x_1)$ by either the local $\Box$-rules or the $\langle U\rangle_+$-rule. In the former case, the only local $\Box$-rules applicable to ψ are the $Ref_{[U]}$-rule or the $Ref_{[D]}$-rule. So, either ψ was obtained from $[U]\psi$ or $[D]\psi$ by applying either the $Ref_{[U]}$-rule or the $Ref_{[D]}$-rule respectively. In both cases, we have that $r([U]\psi) = r([D]\psi) > r(\psi)$, which is impossible. In the latter case, we have $\langle U\rangle\psi \in L(x_0)$, and so $r(x_0) > r(x_1)$ and therefore $r(x_0) > r(x_2)$. ■

Lemma 6.11
Let $n > 0$ and $x_0, x_1, x_2, \ldots, x_n$ be successive nodes such that $\mathcal{L}((x_k, x_{k+1})) = \xi_k$ for $0 \leq k < n$, $\xi_{n-1} \in \{F, P\}$, $\{\xi_k : k < n-1\} \subseteq \{U, D\}$ and $r(x_0) = r(x_n)$. If $[\xi_{n-1}]\psi \in L(x_n)$ and $[\xi_{n-1}]\psi$ is a maximal rank formula in $L(x_n)$, then for all $k \leq n$, we have $[\xi_{n-1}]\psi \in L(x_k)$.

Proof. By downward induction. Let $\xi_{n-1} = F$ (the case where $\xi_{n-1} = P$ is identical). The case where $k = n$ is immediate. So, suppose that it holds for k, we want to show that it holds for $k - 1 \geq 0$. We have that $\xi_{k-1} = D$ or $\xi_{k-1} = U$.

Suppose $\xi_{k-1} = D$ (the case where $\xi_{k-1} = U$ is analogous). We would like to show that $[F]\psi \in L(x_{k-1})$. Since $\xi_{k-1} = D$, we know that $[F]\psi$ was not inserted into $L(x_k)$ by an application of the $[D]_+$-rule or the $\langle D\rangle_+$-rule. So $[F]\psi$ was inserted by an application of the local $\Box$-rules. Here we have three possible cases. (1) $[F]\psi$ was inserted by applying the $Ref_{[U]}$-rule, then $[U][F]\psi \in L(x_k)$. However, $[U][F]\psi$ could not have been inserted by applying the $[D]_+$-rule, and any further application of the $Ref_{[U]}$ rule would increase the rank of the formula, which is impossible. (2) $[F]\psi$ was inserted by applying the $Ref_{[D]}$-rule, then $[D][F]\psi \in L(x_k)$, and $[D][F]\psi$ was inserted by the $[D]_+$-rule to $L(x_{k-1})$. Thus, $[D][F]\psi \in L(x_{k-1})$ and by $Ref_{[D]}$-rule, $[F]\psi \in L(x_{k-1})$. And finally, (3) $[F]\psi$ was inserted by the $Trans_{[F]}$-rule. This means that $\psi = [F]\gamma$. Now $[F]\gamma$ could not have been inserted into $L(x_k)$ by the $[D]_+$-rule, and therefore must have been inserted by a local $\Box$-rule, namely either $Ref_{[U]}$ or $Ref_{[D]}$ rule. If it was inserted by the $Ref_{[U]}$-rule, then by analogous reasoning to case (1), we obtain a contradiction. If $[F]\gamma$ was inserted by the $Ref_{[D]}$-rule, then by analogous reasoning to case (2), we obtain $[F]\gamma \in L(x_{k-1})$ and by applying the $Trans_{[F]}$-rule, we get $[F]\psi \in L(x_{k-1})$. ■

Lemma 6.12
Let $[x_0, x_k)$ be a right-open interval such that $\mathcal{L}((x_0, x_1)) = \xi$ and $\mathcal{L}((x_{k-1}, x_k)) = Inv(\xi)$, for $\xi \in \{F, P\}$ and $\{\mathcal{L}((x_j, x_{j+1})) : 0 < j < k-1\} \subseteq \{U, D\}$. Then $r(x_0) > r(x_k)$.

Proof. Let $\xi = F$ (the case where $\xi = P$ is analogous). Suppose, for proof by contradiction, that $r(x_0) = r(x_k)$. Then by Lemma 6.9 we have $r(x_0) = r(x_j) = r(x_k)$ for $0 \le j \le k$. Suppose ψ is a formula with the maximal rank in $L(x_k)$. Since the application of the local $\Box$-rules do not increase the rank of the formula, we can assume without loss of generality that ψ was inserted into $L(x_k)$ by an application of either the $\langle P\rangle_+$-rule or the $[P]_+$-rule to $L(x_{k-1})$. In the former case we have $\langle P\rangle\psi \in L(x_{k-1})$, and therefore $r(x_0) > r(x_k)$. Contradiction. In the latter case, we have $\psi \in \{[P]\theta, \theta, [D]\theta\}$ and $[P]\theta \in L(x_{k-1})$. From the proof of Lemma 6.9, we can assume $\psi = [P]\theta$.

Since $\{\mathcal{L}((x_j, x_{j+1})) : 0 < j < k-1\} \subseteq \{U, D\}$, it follows from Lemma 6.11 that $\psi \in L(x_1)$.

Now, by supposition $\psi = [P]\theta$ and $\mathcal{L}((x_0, x_1)) = F$. If ψ was inserted into $L(x_1)$ by application of the $[F]_+$-rule, then $[F]\psi$ in $L(x_0)$ and therefore $r(x_0) > r(x_k)$. Contradiction (again). If ψ is the formula that generated the node x_1 then we have $r(x_0) > r(x_1)$, which is impossible. So ψ must have been inserted by the local $\Box$-rules. Here we have two possibilities: (1) ψ was inserted by applying the $Ref_{[U]}$ or $Ref_{[D]}$ rule. Then $L(x_1)$ contains $[U]\psi$ or $[D]\psi$. These in turn must have been inserted by applying either the $[F]_+$ or the $Ref_{[U]}$ or the $Ref_{[D]}$ rule, all of which lead to an increase in the rank of the formula. Contradiction. (2) ψ was inserted by applying the $Trans_{[P]}$-rule, then $\theta = [P]\gamma$. If $[P]\gamma$ was inserted by either the $Ref_{[U]}$ or $Ref_{[D]}$-rule, then we can treat it analogously to case (1) to get the contradiction $r(x_0) > r(x_k)$. Otherwise, $[P]\gamma$ was inserted by application of the $[F]_+$-rule, and so $[F][P]\gamma \in L(x_0)$ and therefore $r(x_0) > r(x_k)$. Contradiction. ■

Henceforth, we will call right-open intervals $[x_0, x_n)$ *short intervals* if they are of the form stated in Lemma 6.10 and Lemma 6.12.

Lemma 6.13

Let $m = |\Phi_\phi|$, $n > m^3$, and ξ be a relation from the set $\{F, P, U, D\}$. Let $[x_0, x_n)$ be a right-open interval of the completion tree with $\mathcal{L}((x_i, x_{i+1})) = \xi$ for $0 \le i < n$. If the $[\xi]_+$-rule cannot be applied to these nodes, then there is a blocked x_i among them.

Proof. First, consider the elements of $B(x_i)$ for $i > 0$. We want to show that

$$A(x_i) \subseteq A(x_{i+1}) \quad \text{for all } 0 < i < n.$$

It is sufficient to show that, if $[\xi]\psi \in L(x_{i-1})$ then $[\xi]\psi \in L(x_i)$. But this is true by application of the $\Box_+$-rule to $L(x_{i-1})$.

Since $A(x_i) \subseteq \Phi_\phi \backslash \{\bot\}$ and $|\Phi_\phi| = m$, we have that

$$|\{A(x_i) : 0 < i \le n\}| \le m$$

This implies (recalling that $B(x_i) = A(x_i) \cup \{C_{x_i}\}$), since we have m choices for C_{x_i},

$$|\{B(x_i) \mid 0 < i \le n\}| \le m^2.$$

Secondly, consider $L(x_i)\backslash Inv(\xi)$. Again, the $\Box_+$-rules yield

$$L(x_i)\backslash Inv(\xi) \subseteq L(x_{i-1})\backslash Inv(\xi) \text{ for all } 1 < i \le n,$$

which implies

$$|\{L(x_i)\backslash Inv(\xi) \mid 1 \le i \le n\}| \le m.$$

Summing up, since $n > m^3$ there must be at least two nodes x_j, x_k (for $1 \leq j < k \leq n$) which satisfy

$$B(x_j) = B(x_k) \quad \text{and} \quad L(x_j)\backslash Inv(\xi) = L(x_k)\backslash Inv(\xi).$$

This implies that one of the nodes is blocked by the other. ■

Lemma 6.14
Let $x_0, x_1, x_2, \ldots$ be successive nodes, preceding away from or towards the root, such that $\mathcal{L}((x_k, x_{k+1})) = \xi_k$ for $k \geq 0$, $\xi_0 \in \{F, P\}$, $Inv(\xi_0) \notin \{\xi_k : k \geq 0\}$, and $\nexists k \geq 0$ such that $\xi_k = U$ and $\xi_{k+1} = D$. Then, if $[\xi_0]\psi \in L(x_0)$ then for all $k \geq 0$, we have

1. $[\xi_0]\psi \in L(x_k)$,
2. if $\xi_k = D$ and $\psi \neq [\xi_0]\beta$ then $[D][\xi_0]\psi \in L(x_k)$,
3. if $\xi_k = D$ and $\psi = [\xi_0]\beta$ then $[D]\psi \in L(x_k)$.

Proof. By induction on k. Let $\xi_0 = F$ (the case where $\xi_0 = P$ is identical). For the base case where $k = 0$, it is immediate that 1, 2 and 3 hold. Now suppose that it is true for k, we want to show that 1, 2 and 3 hold for $k+1$. There are three possibilities as to what ξ_k can be:

1. $\xi_k = F$ and $\psi = [F]\beta$. Then, by the inductive hypothesis, we have $[F]\psi \in L(x_k)$. Therefore, by application of the $[F]_+$-rule, we obtain $\{[F]\psi, \psi, [D]\psi\} \subseteq L(x_{k+1})$.
2. $\xi_k = F$ and $\psi \neq [F]\beta$. Then, by the $Trans_{[F]}$-rule we have $[F][F]\psi \in L(x_k)$ and therefore $\{[F][F]\psi, [F]\psi, [D][F]\psi\} \subseteq L(x_{k+1})$ by the $[F]_+$-rule. By the $Ref_{[D]}$-rule, we have $[F]\psi \in L(x_{k+1})$.
3. $\xi_k = U$. Then by supposition, $\xi_{k+1} \neq D$. Therefore, we only require that $[F]\psi \in L(x_{k+1})$. By the inductive hypothesis, we have $[F]\psi \in L(x_k)$. By application of the $Mon_{[U]}$-rule, we have $[U][F]\psi \in L(x_k)$. By the $[U]_+$-rule, we have $[U][F]\psi \in L(x_{k+1})$, and by the $Ref_{[U]}$-rule, we have $[F]\psi \in L(x_{k+1})$.
4. $\xi_k = D$ and $\psi \neq [F]\beta$. Then, by the inductive hypothesis, we have $[D][F]\psi \in L(x_k)$. By the $[D]_+$-rule, we have $[D][F]\psi \in L(x_{k+1})$, and by the $Ref_{[D]}$-rule, we have $[F]\psi \in L(x_{k+1})$.
5. $\xi_k = D$ and $\psi = [F]\beta$. Then by the inductive hypothesis, we have $[D]\psi \in L(x_k)$. By the $[D]_+$-rule, we have $[D]\psi \in L(x_{k+1})$. By $Ref_{[D]}$-rule, we obtain $\psi \in L(x_{k+1})$ and since $\psi = [F]\beta$, we can apply the $Trans_{[F]}$-rule to get $[F]\psi \in L(x_{k+1})$. ■

Lemma 6.15
Consider any right-open interval $[x_0, x_n)$ not containing any *short intervals*, such that $\mathcal{L}((x_0, x_1)) = \xi$, $\mathcal{L}((x_{n-1}, x_n)) = \xi$ (for $\xi \in \{F, P\}$). Then $A(x_1) \subseteq A(x_n)$ and $L(x_1)\backslash Inv(\xi) \supseteq L(x_n)\backslash Inv(\xi)$.

Proof. As mentioned in the proof of Lemma 6.13, if $[\xi]\psi \in L(x_0)$ then by applying Lemma 6.14, we have $[\xi]\psi \in L(x_{n-1})$, and therefore $A(x_1) \subseteq A(x_n)$. Similarly, by a backwards application of Lemma 6.14, for any $[Inv(\xi)]\theta \in L(x_n)\backslash Inv(\xi)$, we have $[Inv(\xi)]\theta \in L(x_1)\backslash Inv(\xi)$. Thus, we obtain $L(x_1)\backslash Inv(\xi) \supseteq L(x_n)\backslash Inv(\xi)$. ■

Lemma 6.16
Consider any right-open interval $[x_0, x_n)$ not containing any short intervals. If there is no blocked $x_i \in [x_0, x_n)$ then $n \leq 2m^6 + m^3 - 1$.

Proof. If $\mathcal{L}((x_j, x_{j+1})) = \xi_j$ for $0 \leq j < n$, then the sequence of labels $\xi_0, \xi_1, \ldots, \xi_{n-1}$ for a right-open interval with no blocked nodes and not containing any short intervals takes the following regular expression (where * denotes finite repitition and + denotes union):

1. $D^{\leq l}U^{\leq l}((FD^{\leq l}U^{\leq l})^* + (PD^{\leq l}U^{\leq l})^*)$, where $l = m^3$.

Let $\xi \in \{F, P\}$ and let $\{j : 1 \leq j < n, \xi_j \in \{F, P\}\} = \{j_1, \ldots, j_l\}$, where $j_1 < \cdots < j_l$. If $l = 0$ then by (1) it is clear that $n \leq 2m^3 - 2$. Assume then that $l > 0$ and let $\xi_{j_1} = \xi$, then $\xi_{j_k} = \xi$ for all $1 \leq k \leq l$.

To begin with, we have $j_1 \leq 2m^3 - 2$, and furthermore, if $1 \leq k \leq k' \leq l$, we know by Lemma 6.15 above that

$$A(x_{j_k+1}) \subseteq A(x_{j_{k'}+1}) \quad \text{and} \quad L(x_{j_k+1}) \backslash Inv(\xi) \supseteq L(x_{j_{k'}+1}) \backslash Inv(\xi).$$

Now as in Lemma 6.13, we know

$$| \{B(x_{j_k+1}) : 1 \leq k \leq l\} | \leq m^2,$$

and

$$| \{L(x_{j_k+1}) \backslash Inv(\xi) : 1 \leq k \leq l\} | \leq m.$$

So $l \leq m^3$.

Moreover, we also know

$$j_{k+1} - j_k \leq 2m^3 - 1 \quad \text{for all } k \text{ with } 1 \leq k < l.$$

Bringing everything together, we have that the longest possible path before any blocking occurs is $(2m^3 - 2) + m^3(2m^3 - 1) + 1$, which simplifies to $2m^6 + m^3 - 1$. Now, if $n > 2m^6 + m^3 - 1$, there must be at least two nodes x_a, x_b which satisfy

$$B(x_a) = B(x_b) \quad \text{and} \quad L(x_a) \backslash Inv(\xi) = L(x_b) \backslash Inv(\xi).$$

This implies that one of these nodes is blocked by the other, contradiction. ■

We will now use the above lemmas to give a polynomial bound on the length of paths in a completion tree generated by the tableau rules.

Lemma 6.17
The paths of a completion tree for a formula ϕ have a length at most $O(m^7)$ [30] where $m = |\Phi_\phi|$.

Proof. Let $x_0, \ldots, x_n$ be a branch of the completion tree. From Lemma 6.10 and Lemma 6.12, we know that for any short interval $[x_i, x_k)$ on the branch, $r(x_i) > r(x_k)$. Consider a maximal set $\mathcal{D}$ of pairwise disjoint short intervals. The size of $\mathcal{D}$ is bounded by $O(m)$. Furthermore, the length of a short interval of the form in Lemma 6.10 is 2; whilst, the length of a short interval of the form stated in Lemma 6.12 is $O(2m^3 + 1)$.

Now define a *gap* on the branch to be a maximal right-open interval disjoint from $\mathcal{D}$. The number of *gaps* on the branch is $O(1 + m)$. Furthermore, any gap on the branch conforms to the form stated in Lemma 6.16. Its length is then bounded by $O(m^6)$. We therefore conclude that the branch consists of $O(m)$ *short intervals*, each of which has length at most $O(m^3)$, and $O(m)$ *gaps*, each of which has length at most $O(m^6)$. Therefore the length of a path in the completion tree is bounded by $O(m^7)$ or blocking occurs. ■

Theorem 6.18
$\mathrm{SAT}(K_{\mathrm{INT}}) \in \mathrm{PSPACE}$

Proof. We adopt the PSPACE decision procedure presented in [25]. Let $m = |\Phi_\phi|$. For each node x we can store the labels $L(x)$ and $B(x)$ using m bits for each set. We apply the expansion rules as given in Table 1. If a clash is generated, then the current run is rejected. Otherwise we evaluate the completion tree in a depth-first way: we keep track of exactly one path of the completion tree by memorising, for each node x, which of the $\Diamond\psi$ formulas in $L(x)$ have yet to be generated. This can be done using an additional m bits for each node. We have three possible outcomes for an investigation of a subtree below a node x:

- A clash is detected. This implies that the current run is rejected.
- The $\Box_+$-rules lead to an increase of $L(x)$. We reconsider all subtrees below x, re-using the space used for former subtrees below x.
- Neither of the cases above happen. We can then forget about this subtree and start the investigation of another subtree below x. If all subtrees have been investigated, we consider x's predecessor.

Proceeding like this, the algorithm can be implemented using $2m + m$ bits for each node, where the $2m$ bits are used to store the labels of the node, while m bits are used to keep track of the successors already generated. Since we reuse the memory of the successors, we only have to store one path of the completion tree at a time. From Lemma 6.16, the length of this path is bounded by $O(m^7)$. Thus, we can test for the existence of a completion tree using at most $O(m^8)$ bits.

Since we have the $\vee$-rule, we are dealing with a non-deterministic algorithm. However, Savitch's Theorem tells us that there is a deterministic implementation of this algorithm using at most $O(m^{16})$ bits. ■

7 Expressivity of the logic

In this section, we investigate the expressivity of our logic. We highlight its expressive limitations, in particular we identify a number of important modal operators that are not definable in our logic.

7.1 Expressive limitations of the logic

In [43], Venema shows that the $\langle A\rangle$ modality and its converse are definable by the $\langle B\rangle$ and $\langle E\rangle$ operator and their converse. Thus, $\langle B\rangle$ and $\langle E\rangle$ (and their converse) are sufficient to capture all 12 possible relations between two distinct intervals in a linear interval temporal structure (cf. Section 1.1). In this section, we will show that many of these modalities are *not* definable in our logic.

Furthermore, in [26] it was shown that the logic of minimal interval structures in the interval hybrid temporal language (without $\downarrow$) was EXPTIME-complete. The increase in complexity is due to the presence of *nominals* and @ in the interval hybrid temporal language, which allow us to create names for states and therefore reason about state identity. In this section, we show that this mechanism is lacking in the interval temporal language; in particular we show that the *difference* operator is *not* definable in the interval temporal language.

THEOREM 7.1
The $\langle B\rangle$, $\langle E\rangle$ operators (and their converses), and the *difference* operator are not definable in the interval temporal language.

PROOF. The proof is by a couple of simple bisimulation arguments.

First, we show that $\langle B\rangle$ and $\langle E\rangle$ are not definable.

Let $\mathcal{M}_1 = (W^1, R_{<}^1, R_{>}^1, R_{\sqsubseteq}^1, R_{\sqsupseteq}^1, V^1)$ and $\mathcal{M}_2 = (W^2, R_{<}^2, R_{>}^2, R_{\sqsubseteq}^2, R_{\sqsupseteq}^2, V^2)$, where $W^1 = \{[2,10],[2,4],[7,14],[12,14]\}$ and $W^2 = \{[3,8],[10,14]\}$ of closed intervals over the rationals. The relations on $\mathcal{M}_1$ are defined as follows:

1. $R_{<}^1 = \{([2,10],[12,14]),([2,4],[7,14]),([2,4],[12,14])\} = (R_{>}^1)^{\smile}$,
2. $R_{\sqsubseteq}^1 = \{([x,y],[x,y]) : [x,y] \in W^1\} \cup \{([2,4],[2,10]),([12,14],[7,14])\} = (R_{\sqsupseteq}^1)^{\smile}$.

The relations on $\mathcal{M}_2$ are defined as follows:

1. $R_{<}^1 = \{([3,8],[10,14])\} = (R_{>}^1)^{\smile}$,
2. $R_{\sqsubseteq}^1 = \{([3,8],[3,8]),([10,14],[10,14])\} = (R_{\sqsupseteq}^1)^{\smile}$.

It can be easily verified that both structures are minimal interval structures. A simple bisimulation argument suffices to show that these two structures are indistinguishable.

Let $Z = \{([2,4],[3,8]),([2,10],[3,8]),([7,14],[10,14]),(12,14],[10,14])\}$ be a relation between $\mathcal{M}_1$ and $\mathcal{M}_2$. Let $V^1(p) = W^1$ and $V^2(p) = W^2$. Recall that a bisimulation between $\mathcal{M}_1$ and $\mathcal{M}_2$ is a non-empty relation $Z \subseteq W^1 \times W^2$ such that:

1. If aZa' then a and a' satisfy the same proposition letters.
2. If aZa' and aR^1c (for $R \in \{R_{<}, R_{>}, R_{\sqsubseteq}, R_{\sqsupseteq}\}$), then there exists c' (in $\mathcal{M}_2$) such that cZc' and $a'R^2c'$ (the *forth* condition).
3. The converse of 2.

That Z is a bisimulation can be easily verified. Thus, $\mathcal{M}_1$ and $\mathcal{M}_2$ are bisimilar. Now, it is clearly the case that $\mathcal{M}_1,[2,10],V^1 \models \langle B\rangle p$; whilst $\mathcal{M}_2,[3,8],V^2 \not\models \langle B\rangle p$. Since modal formulas are invariant under bisimulations [11], we conclude that the $\langle B\rangle$ operator is not definable in our logic. It is also the case that $\mathcal{M}_1,[7,14],V^1 \models \langle E\rangle p$; whilst $\mathcal{M}_2,[10,14],V^2 \not\models \langle E\rangle p$. Therefore, we conclude that the $\langle E\rangle$ operator is not definable in our logic. The case of the converse modalities, $\langle \overline{B}\rangle$ and $\langle \overline{E}\rangle$, are treated analogously.

Now, we show that the *difference* operator is not definable. Before proceeding any further, we will define the difference operator (D). For any model $\mathcal{M}$, D must be interpreted using the inequality relation $\neq$. That is:

$$\mathcal{M}, w \models D\phi \quad \text{iff } \exists u(u \neq w \wedge \mathcal{M}, u \models \phi)$$

Now let $\mathcal{M}_1 = (W^1, R_{<}^1, R_{>}^1, R_{\sqsubseteq}^1, R_{\sqsupseteq}^1, V^1)$ and $\mathcal{M}_2 = (W^2, R_{<}^2, R_{>}^2, R_{\sqsubseteq}^2, R_{\sqsupseteq}^2, V^2)$, where $W^1 = \{[2,10],[7,10],[7,14]\}$ and $W^2 = \{[4,8]\}$ of closed intervals over the rationals. The relations on $\mathcal{M}_1$ are defined as follows:

1. $R_{<}^1 = \emptyset = (R_{>}^1)^{\smile}$,
2. $R_{\sqsubseteq}^1 = \{([x,y],[x,y]) : [x,y] \in W^1\} \cup \{([7,10],[2,10]),([7,10],[7,14])\} = (R_{\sqsupseteq}^1)^{\smile}$.

The relations on $\mathcal{M}_2$ are defined as follows:

1. $R^1_< = \emptyset = (R^1_>)^\smile$,
2. $R^1_\sqsubseteq = \{([4, 8], [4, 8])\} = (R^1_\sqsupseteq)^\smile$.

It can be easily verified that both structures are minimal interval structures. Now let $Z = W^1 \times W^2$. Let $V^1(p) = W^1$ and $V^2(p) = W^2$. That Z is a bisimulation can be easily verified. It can then be easily seen that $\mathcal{M}_1, [2, 10], V^1 \models Dp$; whilst $\mathcal{M}_2, [4, 8], V^2 \not\models Dp$. We therefore conclude that the *difference* operator is also not definable in our logic.

8 Concluding remarks

In [23] it was shown that the satisfiability problem for interval structures depended critically on our underlying assumptions about time. For many interesting classes of structures, the satisfiability problem is undecidable. In our article, we introduce an interval logic, with a simple syntax and semantics, whose computational complexity is considerably lower. We show that the logic for the general class of minimal interval structures has the finite model property and is decidable, and that the satisfiability problem is PSPACE-complete. We go on to identify some important limitations in its expressivity. We should add that ours is not the only approach to obtaining decidable interval temporal logics, [19] gives a recent survey of results on interval temporal logics. In addition, it was shown in [29] that even over the class of substructures of interval structures over the reals, the resulting logic is undecidable. This suggests that the results obtained in this article are already very close to the border between decidable and undecidable.

A couple of interesting open questions arise from our enterprise:

1. What, if any, further assumptions are needed to obtain concrete interval temporal structures? And what further assumptions about time (such as convexity, linearity, density etc.) can we make whose addition/inclusion makes the satisfiability problem (un)decidable?
2. Which, if any, other modalities can we incorporate into our language without losing decidability? Can some of the undefinable operators (like the difference operator) be added to our logic whilst preserving decidability and complexity?

Acknowledgements

I am grateful to Ian Hodkinson and Maarten Marx for their assistance and guidance throughout. Also, I would like to thank Patrick Blackburn for his advice and encouragement, and Jorge Petrucio Viana and Nick Bezshanishvili for helpful discussions on various parts of the article. This study is supported by EPSRC Grant No. GR/S19004/04. Finally, I would like to express my gratitude to the anonymous reviewers whose comments helped to improve the article.

References

[1] J. Allen and J. Koomen. Planning using a temporal world model. In *Proc. 8th IJCAI*, pp. 741–747, Morgan Kaufmann, 1983.

[2] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, **23**, 123–154, 1984.

[3] J. F. Allen. *Reasoning about Plans*. Morgan Kaufmann, 1991.
[4] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, **8**, 653–679, 2000.
[5] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: characterization, interpolation and complexity. *Journal of Symbolic Logic*, **66**, 977–1010, 2001.
[6] F. Baader, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, **1**, 109–138, 1993.
[7] F. Baader and H. J. Ohlbach. A multi-dimensional terminological knowledge representation language. *Journal of Applied Non-Classical Logics*, **5**, 153–197, 1995.
[8] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, **10**, 137–168, 2000.
[9] P. Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL*, **8**, 339–365, 2000.
[10] P. Blackburn and B. ten Cate. Beyond pure axioms: node creating rules in hybrid tableaux. In C. Areces et al., eds, *Proceedings of the 4th Workshop on Hybrid Logics (Hylo 2002)*.
[11] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2002.
[12] P. Blackburn and J. Seligman. Hybrid Languages, *Journal of Logic, Language and Information*, **4**, 251–272, 1995.
[13] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*, MIT Press, 1995.
[14] M. Finger, D. Gabbay, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, Vol. 2, Oxford University Press, 2000.
[15] D. Gabbay and V. Shehtman. Products of modal logics. Part I. *Journal of the IGPL*, **6**, 73–146, 1998.
[16] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional Modal Logics: Theory and Applications*, Studies in Logic, Vol. 148, Elsevier, 2003.
[17] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Products of 'transitive' modal logics. *Journal of Symbolic Logic*, **70**, 993–1021, 2005.
[18] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Non-primitive recursive decidability of products of modal logics with expanding domains. *Annals of Pure and Applied Logic*, **142**, 245–268, 2006.
[19] V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics. *Journal of Universal Computer Science, Special Issue on Spatial and Temporal Reasoning*, **9**, 1137–1167, 2003.
[20] J. Halpern, Z. Manna, and B. Moszkowski. A high-level semantics based on interval logic. In *Proceedings of the 10th ICALP*, pp. 278–291, Springer Verlag, 1983.
[21] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the Association for Computing Machinery*, **38**, 935–962, 1991.
[22] D. Harel, D. Kozen, and R. Parikh. Process logic: expressiveness, decidability, completeness. *J. Comput. Syst. Sci.*, **25**, 145–180, 1982.
[23] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *Proceedings of the IEEE Symposiun on Logic in Computer Science*, pp. 279–292, 1986.
[24] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the Association for Computing Machinery*, **38**, 935–962, 1991.

[25] I. Horrocks, U. Sattler, and S. Tobies. A PSPACE-algorithm for deciding $ALCNI_{R^+}$-satisfiability. *LTCS-Report 98–08*, *LuFg Theoretical Computer Science*, RWTH Aachen, Germany, 1998.
[26] A. Hussain. A minimal hybrid logic for intervals. *Logic Journal of the IGPL*, **14**, 35–62, 2006.
[27] R. A. Kowalski and M. J. Sergot. A logic-based calculus of events. *New Generation Computing*, **4**, 67–96, 1986.
[28] R. Ladner. The computational complexity of provability in systems of modal logic, *SIAM Journal of Computing* **6**, 467–480, 1977.
[29] C. Lutz and F. Wolter. Modal logics of topological relations. *Logical Methods in Computer Science*, 2006.
[30] C. Papadimitriou. *Computational Complexity*, Addison Wesley, 1994.
[31] R. Parikh. A decidability result for second order process logic. In *Proc. of the 19th Annual Symposium on Foundations of Computer Science*, IEEE, pp. 177–183, 1978.
[32] V. R. Pratt. Process logic. In *Proc. of the 6th Annual Symposium on the Principles of Programming Languages*, pp. 93–100, 1979.
[33] A. Pnueli. A temporal logic of programs. In *Proc. of the 18th Annual Symposium on Foundations of Computer Science*, IEEE, pp. 46–57, 1977.
[34] A. N. Prior. *Past, Present and Future*. Clarendon Press, Oxford, England, 1967.
[35] V. R. Pratt. Models of program logics. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pp. 115–122, 1979.
[36] M. Reynolds. A decidable temporal logic of parallelism. *Notre Dame Journal of Formal Logic*, **38**, 419–436, 1997.
[37] M. Reynolds and M. Zakharyashev. On the products of linear modal logics. *Journal of Logic and Computation*, **11**, 909–931, 2001.
[38] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, **2**, 77–96, 1973.
[39] V. Shehtman. Two-dimensional modal logics. *Mathematical Notices of the USSR Academy of Sciences*, **23**, 417–424, 1978. (Translated from Russian)
[40] E. Spaan. *Complexity of modal logics*. PhD thesis, ILLC. University of Amsterdam, 1993.
[41] E. Spaan. The complexity of propositional tense logics. In M. de Rijke, ed., *Diamonds and Defaults: Studies in Pure and Applied Intensional Logic*, Kluwer Academic Publishers, pp. 287–307, 1993.
[42] J. van Benthem. *The Logic of Time*. 2nd edn, Kluwer Academic Publishers, Dordrecht 1991.
[43] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, **31**, 529–547, 1990.
[44] Y. Venema. Temporal logic. In L. Goble, ed., *Blackwell Guide to Philosophical Logic*, Blackwell Publishers, pp. 203–223, 1998.
[45] Y. Venema. Points, lines and diamonds: a two-sorted modal logic for projective planes. *Journal of Logic and Computation*, **9**, 601–621, 1999.
[46] F. Wolter and M. Zakharyaschev. Qualitative spatio-temporal representation and reasoning: a computational perspective. In G. Lakemeyer and B. Nebel, eds, *Exploring Artificial Intelligence in the New Millenium*, pp. 175–216, Morgan Kauffman, 2002.

Received 1 July 2005