J Log Lang Inf (2011) 20:49–68
DOI 10.1007/s10849-010-9126-5

# An Event-Based Fragment of First-Order Logic over Intervals

**Savas Konur**

Published online: 22 July 2010

**Abstract** We consider a new fragment of first-order logic with two variables. This logic is defined over interval structures. It constitutes unary predicates, a binary predicate and a function symbol. Considering such a fragment of first-order logic is motivated by defining a general framework for event-based interval temporal logics. In this paper, we present a sound, complete and terminating decision procedure for this logic. We show that the logic is decidable, and provide a NEXPTIME complexity bound for satisfiability. This result shows that even a simple decidable fragment of first-order logic has NEXPTIME complexity.

**Keywords** Two variable fragments of first-order logic · Interval temporal logics · Decidability · Tableau methods

## 1 Introduction

Propositional interval logics are very expressive temporal logics, with simple, syntax and semantics, which allow one to naturally express statements that refer to time intervals. They provide a natural framework for temporal representation and reasoning. However, many of these logics usually exhibit a bad computational behaviour, and they are undecidable in most of the cases. The main species of studied propositional interval temporal logics include Moszkowski's Propositional Interval Logic (PITL) (Moszkowski (1983)), Halpern and Shoham's modal logic of time intervals (HS) (Halpern and Shoham 1991), Venema's CDT logic (Venema (1991)) (extended to branching-time frames with linear intervals by Goranko et al. 2006), and Montanari, Goranko and Sciavicco's Propositional Neighborhood Logics (PNL) (Goranko 2003).

S. Konur (✉)
Department of Computer Science, The University of Liverpool, Liverpool, L69 3BX, UK
e-mail: konur@liverpool.ac.uk

In some cases, the full expressive power of interval logics might not be needed. In such situations, decidability can be obtained through some restrictions. For example, there are some contexts where interpretations in which infinitely many statements (events) hold (occur) in a finite space of time are of no interest. Examples can be found in computational linguistics. Pratt-Hartmann (2005) and Konur (2008) developed decidable interval logics of temporal prepositions which are interpreted over finite models. These logics are convenient for expressing the semantics of natural language constructions, and for specifying event-based real-time system requirements. One important aspects of these logics is that they are genuinely interval-based, and they do not impose semantic restrictions, such as locality.

Since the logics defined in (Pratt-Hartmann 2005; Konur 2008) are modal logics, a first-order logic can defined for these logics (and similar types of logics) as a general framework. In this paper, we define such a framework by studying a *new* two-variable fragment of first-order logic where unary predicates represent event types, the only binary predicate represents an interval relation, and the only function symbol represents the 'duration' operation. We call this new logic EF, which is defined over *interval* structures. Although it is a simple logic, its genuinely new syntax makes this logic worth to be explored. Although well-known propositional interval logics, like HS, CDT, PNL and PITL are very expressive, it can be easily shown that EF cannot be reduced to these logics (A theoretical analysis of comparing EF to these logics is outside the scope of this paper).

By studying the logic EF, we are able to investigate a new decidable fragment of first-order logic (FOL). In the literature, there are various decidable fragments of FOL. Mortimer (1975) showed that the two-variable fragment of first-order logic ($FO^2$) has the finite model property, and hence decidable for satisfiability. One of the reasons for the significance of this result is that many propositional modal logics can be embedded into $FO^2$ (Grädel and Otto 1999). Recently, the bound on model size has been improved to locate the complexity of the satisfiability problem for $FO^2$ in NEXPTIME-complete (Grädel et al. 1997).

In Otto (2001) the satisfiability problem for $FO^2$ is investigated over finite and infinite linearly ordered and well-ordered domains, as well as over finite and infinite domains in which one or several designated binary predicates are interpreted as arbitrary well-founded relations. It has been shown that $FO^2$ over ordered and well-ordered domains or in the presence of one well-founded relation, is decidable for satisfiability and for finite satisfiability. Actually, the complexity of these decision problems is NEXPTIME. In contrast, $FO^2$ becomes undecidable for satisfiability and for finite satisfiability, if a sufficiently large number of predicates is required to be interpreted as orderings, well-orderings, or as arbitrary well-founded relations. This undecidability result also entails the undecidability of the natural common extension of $FO^2$.

Andreka et al. (1996) defined the *guarded fragment of first-order logic (GF)*. The authors dropped the restriction to use only two variables and only monadic and binary predicates, but insisted that all quantifiers must be relativized (or 'guarded') by atomic formulas. GF is interesting because it extends many propositional modal logics, because it has useful model-theoretic properties and especially because it is a decidable class that avoids the usual syntactic restrictions (on the arity of relation symbols, the quantifier pattern or the number of variables) of almost all other known decidable fragments of first-order logic. GF also has the finite model property, i.e. every satisfiable formula in the guarded fragment also has a finite model.

In van Benthem (1997) the guarded fragment is generalized to the *loosely guarded fragment (LGF)* where quantifiers are guarded by conjunctions of atomic formulae of certain forms. The loosely guarded fragment has very similar properties of the guarded fragment.

In Grädel (1999), the computational complexity of both guarded and loosely guarded fragments is investigated. It is proved that the satisfiability problems for the guarded fragment (GF) and the loosely guarded fragment (LGF) of first-order logic are complete for deterministic double exponential time. For the subfragments that have only a bounded number of variables or only relation symbols of bounded arity, satisfiability is EXPTIME-complete. Grädel (1999) further establishes a tree model property for both the guarded fragment and the loosely guarded fragment, and gives a proof of the finite model property of the guarded fragment. It is also shown that some natural, modest extensions of the guarded fragments are undecidable.

Although the fragments of FOL mentioned above are considered quite expressive logics and the syntax of the logic EF is relatively simple, the expressive power of EF is not comparable with the expressive power of these logics. Indeed, EF formulas cannot be reduced to these fragments, which makes EF original, and worth exploring.

An important point is that EF is interpreted over interval structures endowed with subinterval relations. Actually, logics of subinterval relations have been studied very little yet. The study of subinterval structures and logics turns out to be important because they occupy a region on the very borderline between decidability and undecidability, and since decidability results in that area are preciously scarce, complete and terminating tableau systems like those constructed in the paper are of particular interest.

In this paper, we propose a terminating tableau system for EF, thus showing that its satisfiability problem is decidable. We, indeed, provide a complexity bound for satisfiability, showing that this problem can be solved in NEXPTIME. This results shows that even a simple decidable fragment of first-order logic has NEXPTIME complexity.

The rest of the paper is organized as follows: In Sect. 2 we define syntax and semantics of the logic EF. In Sect. 3 we show how we construct models, and determine a limit on the size of satisfying models. In Sect. 4 we propose a terminating tableau system for the logic EF, and show that its satisfiability problem is decidable. We conclude in Sect. 5 we with some future research directions.

## 2 The Logic EF

Since the logic EF is a two-variable logic, its formulas contain only two variable symbols, which range over intervals. In the rest of this paper we take an interval to be a closed, bounded and non-empty subset of the real line. More formally we say that an *interval* is a pair $[t_1, t_2]$ such that $t_1, t_2 \in \mathbb{R}$ and $t_1 \leq t_2$. We denote the set of all intervals $\{[t_1, t_2] : t_1 \leq t_2 \wedge t_1, t_2 \in \mathbb{R}\}$ by $\mathcal{I}$, and we use letters $I, J, \ldots$, as intervals. It can be simply observed that intervals may be punctual. Note that due to underlying temporal structure, time in EF is continuous, linear and complete.

Another feature of EF is that it is interpreted over a linear time flow with only finitely many events able to occur over a bounded-time interval. EF formulas are evaluated relative to time-intervals. Event-types are represented by predicate symbols with arity one (unary predicate symbols). Having event-types in the syntax of the language allows us to formalize event-based sentences of a natural language and event-based system specifications of a real-time system. EF also incorporates the notion of *duration* (of an event).

It is also important to mention that we impose some restrictions on the syntax of EF formulas. One restriction is that we only allow unary predicate symbols, and there is only one binary predicate symbol, which is $S$. We do not allow any predicate symbol whose arity is greater than two. In addition, there is only one unary function symbol, which is $\ell$. We do not allow any function symbol with arity greater than one. Having these restrictions, formulas of the logic EF are constructed from the following set of symbols:

- a finite set of *temporal variables*
- a finite set of *predicate* symbols
- a *function* symbol
- a finite set of *operators*: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \bot, \top, \exists, \forall, =, <, >, \leq, \geq$.
- a finite set of *auxiliary symbols*: parentheses, comma.
- a countable sets of *constant symbols.*

Before giving the syntax of the logic EF, we will discuss the following remarks: First, temporal variables, denoted $x, y, z, \ldots$, range over intervals. Second, unary predicate symbols, denoted $e_1, e_2, e_3, \ldots$, represent event-types. We interpret any unary predicate $e$ so that it is satisfied by all and only those time intervals over which $e$ occurs. From now on, we will treat a unary predicate $e$ as an event atom. We will think of $e(J)$ as the occurrence of $e$ over $J$ (where $J$ is an interval). Third, the binary predicate symbol $S$ denotes *the (non-strict) subinterval relation*, which is defined as follows: $[t_1, t_2]$ is a non-strict subinterval of $[t_3, t_4]$ iff $t_1 \subseteq t_3$ and $t_2 \subseteq t_4$. Finally, the unary function symbol $\ell$ denotes the length function which returns the length of an interval.

In the sequel, let $\mathcal{E}$ be a finite set. We refer to elements of $\mathcal{E}$ as *event atoms.*

**Definition 1** Let $e \in \mathcal{E}$ be an event atom, $S$ be a predicate symbol, $\ell$ be a function symbol, $x, y$ be temporal variables, $k$ be a constant, $\psi$ be an EF formula, and $\tau \in \{<, \leq, =, \geq, >\}$. The logic EF is defined by induction as follows:

- $\top$ and $\bot$ belong to EF;
- The following formulas belong to EF: $\exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$ $\forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$;
- EF is closed under Boolean connectives $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$.

We assume that a function $\overline{S} \in \mathcal{I}^2 \rightarrow \{\top, \bot\}$ is associated with the predicate symbol $S$, and a function $\overline{\ell} \in \mathcal{I} \rightarrow \mathbb{R}$ is associated with the function symbol $\ell$.

As for the semantics, assume $J$ is a witness for the temporal variable $x$ in Definition 1, and the free variable $y$ is assigned to $I.e(J)$ means that $\langle J, e\rangle$ is an entry in an

EF model $\mathcal{M}$ (see Definition 2). In other words, $e(J)$ returns true if $\langle J, e\rangle \in \mathcal{M}$, and false otherwise. $\overline{S}(J, I)$ returns true if $J$ is a (non-strict) subinterval of $I$ (i.e. $J \subseteq I$), and returns false otherwise. Finally, $\overline{\ell}(I)\tau k$ returns true if $|I|\ \tau k$, and false otherwise (where $|I|$ denotes the length of the interval $I$).

One important characteristic of EF formulas is the 'quasi-guarded' nature of the quantification they feature. Thus, for example, the formula $\exists x(e(x) \wedge S(x, y) \wedge \ell(x) \geq 0)$ existentially quantifies over intervals satisfying the predicate $e$ ( Similarly for universal formulas). So it does not quantify over all subintervals of the current interval of evaluation without restriction. However, many modal logics, such as HS and CDT, lack the 'quasi-guarded' character of the quantification that EF formulas feature. This feature is very important to have a computationally manageable logic.

Before ending this section, we give an example of representing the meaning of a sentence in EF. Consider the sentence "An alarm was sounded", which asserts that within the given temporal context, there is an interval over which an alarm was sounded. Interpreting unary predicate $alarm$ so that it is satisfied by all and only those time intervals over which an alarm was sounded, we may thus represent the meaning of this sentence by the formula $\exists x(alarm(x) \wedge S(x, y) \wedge \ell(x) \geq 0)$. Note that the temporal context to which the quantification in the sentence above is limited is represented by the free variable $y$ (which is mapped to an interval).

When we introduce an EF formula using the notation $\varphi(x)$, we mean that $x$ occurs free in $\varphi$. Similarly, we use the notation $\varphi(y)$. When a formula has been introduced as $\varphi(x)$, and we later on write $\varphi(y)$, then this formula stands for the formulas which is obtained from $\varphi$ by exchanging $x$ and $y$. Symmetrically, when a formula has been introduced as $\varphi(y)$ and we later on write $\varphi(x)$, we mean the formula which is obtained from $\varphi$ by exchanging $x$ and $y$.

## 3 Building Models

In this section we show that the depth of an EF model can be polynomially bounded by the length of a given formula $\varphi$ whose satisfiability is checked. We prove this by finding a reduced satisfying model, whose depth is bounded by $|\varphi|^2$. This result is important in determining a limit on the size of a satisfying model.

**Definition 2** Let $\mathcal{I}$ be the set of all bounded, closed and non-empty intervals of real numbers, and $\mathcal{E}$ be a finite set of event atoms. An EF **model** $\mathcal{M}$ is a finite subset of $\mathcal{I} \times \mathcal{E}$. For any $J \in \mathcal{I}$ and $e \in \mathcal{E}$, $\mathcal{M}(J)$ and $\mathcal{M}(e)$ are defined as follows:

$$\mathcal{M}(e) \equiv \{J \in \mathcal{I} \mid \langle J, e\rangle \in \mathcal{M}\}$$
$$\mathcal{M}(J) \equiv \{e \in \mathcal{E} \mid \langle J, e\rangle \in \mathcal{M}\}$$

As can be seen from the construction an EF model, intervals are primitive objects of the model. Given that $\varphi$ is an EF formula with one free variable, $\mathcal{M}$ is an EF model, and $I$ is an interval, we write $\mathcal{M} \models \varphi[I]$ if $\varphi$ holds in $\mathcal{M}$ with respect to the variable assignment that maps the free variable to $I$. Given two EF formulas $\varphi(x)$ and $\varphi'(x)$, we say that $\varphi(x) entails \varphi'(x)$ if for all $\mathcal{M}$ and $I$, $\mathcal{M} \models \varphi[I]$ implies $\mathcal{M} \models \varphi'[I].\varphi(x)$

and $\varphi'(x)$ are *logically equivalent* if $\varphi(x)$ entails $\varphi'(x)$ and $\varphi'(x)$ entails $\varphi(x)$. Given a set of formulas $\Phi$, we write $\mathcal{M} \models \Phi[I]$ if $\mathcal{M} \models \varphi[I]$ for all $\varphi \in \Phi$. $\Phi$ is *satisfiable* if for some $\mathcal{M}$ and $I$, $\mathcal{M} \models \Phi[I]$.

We remark that the condition in the above Definition 2 that models are finite subsets of $\mathcal{I} \times \mathcal{E}$ is significant. Because there might be some EF formulas which cannot be satisfied in a finite model. Consider, for example, the following formula:

$$\begin{aligned}&\exists x(e(x) \wedge S(x, y) \wedge \ell(x) \geq 0) \wedge \\ &\forall x(e(x) \wedge S(x, y) \wedge \ell(x) \geq 0 \rightarrow \exists x'(e(x') \wedge S(x', x) \wedge \ell(x') \geq 0)).\end{aligned}$$

This formula is not satisfiable in a finite model; because it implies that every occurrence of $e$ over an interval $J$ requires another $e$ to occur over a subinterval of $J$. Therefore, the formula is unsatisfiable in a finite model.

After saying that EF formulas can only be satisfied in a finite model, we now turn to determining a limit on the size of such a model. In fact, in the next section we will establish an exponential bound on the size of satisfying models. Below we will prove that a satisfying model has a polynomial depth bound on the size of the formula; but before that, we will show how to normalize an EF formula to the desired form.

**Lemma 1** *Every EF formula is logically equivalent to one in which* $\neg$ *appears only in subformula of the form* $\bot (= \neg\top)$.

*Proof* The proof is trivial for $\bot$ . In an EF formula $\neg$ can be moved inwards as follows:

$$\begin{aligned}\neg\exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x)) &\equiv \forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \neg\psi(x)) \\ \neg\forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x)) &\equiv \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \neg\psi(x))\end{aligned}$$

where $\tau \in \{<, \leq, =, \geq, >\}$ and $\tau'$ is the corresponding inverted operator of $\tau$.[1] □

By means of Lemma we can normalize the forms of EF formulas.

**Definition 3** Given an EF formula $\varphi$ and a non-empty model $\mathcal{M}$, the **depth** of $\mathcal{M}$ is the greatest $m$ for which there exist $J_1 \subseteq \cdots \subseteq J_m$ such that for all $i$, $1 \leq i \leq m$ and for some $e \in \mathcal{E}$, $\langle J_i, e\rangle \in \mathcal{M}$. The depth of an empty model is defined to be 0.

Now we will show that the depth of models can be polynomially bounded by the length of the formula. The proof relies on finding a reduced satisfying model $\mathcal{M}^* \subseteq \mathcal{M}$, whose depth is bounded by $|\varphi|^2$, such that $\mathcal{M} \models \varphi[I]$ implies $\mathcal{M}^* \models \varphi[I]$ for a given interval $I$. Before starting the formal proof, we will give some definitions.

**Definition 4** Let $\varphi$ be an EF formula which has the form guaranteed by Lemma 1, $e$ be an event atom, and $J \in \mathcal{I}$ be an interval. Assume that $\mathcal{M}$ contains only event atoms involved in $\varphi$. We define $L_e\,(J)$as follows:

$$\begin{aligned}&L\,(J) = \{\psi(x) \mid \psi(x) \text{is a subformula of } \varphi \text{ with one free variable s.t. } \mathcal{M} \models \psi[J]\} \\ &L_e\,(J) = L\,(J) \setminus \bigcup \{L\,(K) \mid K \subseteq J,\ \langle K, e\rangle \in \mathcal{M}\}\end{aligned}$$

[1] For example, $\leq$ is the corresponding inverted operator of $>$.

$L(J)$ records the subformulas of $\varphi$ which are true at an interval $J$. If we look at the definition, we can see that $L_e(J)$ records the subformulas of $\varphi$ which are true at an interval $J$, except the subformulas which are true at some subinterval $K$ of $J$ with $\langle K, e\rangle \in \mathcal{M}$. We say that a pair $\langle J, e\rangle \in \mathcal{M}$ is *redundant* if $L_e(J) = \emptyset$.

**Lemma 2** *Let the number of symbols in a given EF formula $\varphi$ be denoted by $|\varphi|$. For a given model $\mathcal{M}$, and interval $I$, if $\mathcal{M} \models \varphi[I]$, then there exists a model $\mathcal{M}^* \subseteq \mathcal{M}$, with depth at most $O\left(|\varphi|^2\right)$, such that $\mathcal{M}^* \models \varphi[I]$.*

*Proof* Assume that $\varphi$ has the form guaranteed by Lemma 1. Now we will reduce the model $\mathcal{M}$ to $\mathcal{M}^*$ by removing redundant pairs:

$$\mathcal{M}^* = \mathcal{M} \setminus \{\langle J, e\rangle \mid \langle J, e\rangle \ \ is\ redundant\}$$

Let $m$ be the number of event atoms occurring in $\varphi$, and $n$ be the number of subformulas of $\varphi$. If $J \subseteq J'$ such that $\langle J, e\rangle \in \mathcal{M}$ and $\left\langle J', e\right\rangle \in \mathcal{M}$, then $L_e(J)$ and $L_e\left(J'\right)$ are disjoint. That is, the length of a chain of the intervals at which e occurs is bounded by the number of the subformulas of $\varphi$ in which $e$ is mentioned. Therefore, $\mathcal{M}^*$ is bounded by $m\,(n+2)$. Since we know that $m < |\varphi|$ and $n < |\varphi|$, it easily follows that the depth of $\mathcal{M}^*$ is bounded by $|\varphi|^2$.

Now by using structural induction on the complexity of $\varphi$ we will show that for every interval $I$ and every subformula $\xi$ of $\varphi$, $\mathcal{M} \models \xi[I]$ implies $\mathcal{M}^* \models \xi[I]$.

***Base Case*** **:**

Suppose $\mathcal{M} \models \xi[I]$

$\xi = \top$ or $\xi = \bot$: Trivial

***Inductive Case*:**

Suppose $\mathcal{M} \models \xi[I]$

$\xi = \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$ : Let $J$ be a witness for the existential quantifier in $\xi$, where $y$ takes the value $I$. By the semantics, $\langle J, e\rangle \in \mathcal{M}$ such that $J \subseteq I$, $\ell(J)\tau k$ and $\mathcal{M} \models \psi[J]$. We choose such a $J$ which is minimal under the order $\subseteq$, so that $\langle J, e\rangle \in \mathcal{M}^*$. By the inductive hypothesis, $\mathcal{M}^* \models \psi[J]$. We now have $\langle J, e\rangle \in \mathcal{M}^*$, $J \subseteq I$, $\ell(J)\tau k$ and $\mathcal{M}^* \models \psi[J]$. Thus, $\mathcal{M}^* \models \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$.

$\xi = \forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$ : Let $y$ be mapped to the interval $I$. By the semantics, for every witness $J$ of $x$ $\langle J, e\rangle \in \mathcal{M}$, $J \subseteq I$ and $\ell(J)\tau k$ imply $\mathcal{M} \models \psi(J)$. By construction, $\mathcal{M}^* \subseteq \mathcal{M}$. Since $\xi$ is satisfied by $\mathcal{M}$, it has to be satisfied by its subset $\mathcal{M}^*$. By the inductive hypothesis, $\mathcal{M}^* \models \psi(J)$ for every witness $J$. Thus, $\mathcal{M}^* \models \forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$. □

Lemma 2 shows that, in determining satisfiability of EF formulas, we need never consider very deep interpretations. We now illustrate the basic idea with an example. Assume $I_1, I_2, I_3$ are intervals with $I_3 \subseteq I_2 \subseteq I_1$, and $\mathcal{M}$ is the model $\{\langle I_1, e\rangle\, \langle I_2, e\rangle, \langle I_3, e\rangle\}$, as shown in Part *(i)* of Fig. 1. Let $\phi = \exists x(e(x) \wedge S(x, y) \wedge \ell(x) \geq 0 \wedge (\exists x'(e(x') \wedge S(x', x) \wedge \ell(x') \geq 0))$. Obviously, for any $I \supseteq I_1$, $\mathcal{M} \models \phi\,[I]$. However, it is clear that we can remove the occurrence of $e$ at $I_3$ (alternatively, $I_1$ or $I_2$) without compromising this fact. Thus, if $\mathcal{M}^*$ is the model $\{\langle I_1, e\rangle\, \langle I_2, e\rangle\}$ depicted in Part *(ii)* of Fig. 1, we still have, for any $I \supseteq I_1$, $\mathcal{M}^* \models \phi\,[I]$.

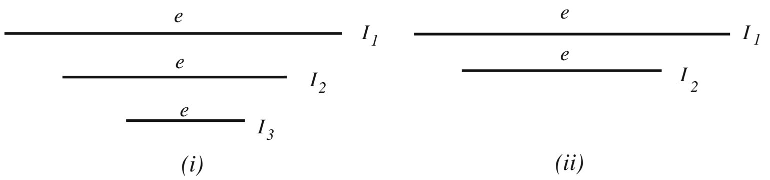

**Fig. 1** Two models making $\phi$ true at any $I \supseteq I_1$

We have shown that the depth of a satisfying model is bounded by $|\varphi|^2$. In the next section we will show that the size of this model is bounded by $2^{p(|\varphi|)}$ for some fixed polynomial $p$. We will actually derive the model from the tableau generated by a tableau procedure.

## 4 A Tableau Based Decision Procedure for EF

In this section we propose a terminating tableau system for the logic EF, thus showing that its satisfiability problem is decidable. Indeed, the satisfiability problem for EF is in NEXPTIME. This is proved by building models whose sizes are exponentially bounded.

In the following, we define a tableau-based decision procedure for EF, and analyze its computational complexity. Then, we prove its soundness and completeness. The procedure is based on an *expansion strategy*. The expansion strategy involves three rules: the *interval relation rule,* which nondeterministically guesses the interval relation among nodes in the graph, the *existential node expansion rule*, which expands existential subformulas in a node and the *universal node expansion rule*, which expands universal subformulas in a node. A blocking condition guarantees the termination of the method.

### 4.1 Preliminary Notions

In the following we introduce some preliminary notions which will be used throughout the rest of the paper.

**Definition 5** A **successor** of a node $v$ is a node $w$ such that there is an edge from $v$ to $w$. A **path** is a sequence of nodes $v_1, \ldots, v_k$ such that for all $1 \leq i < k$, $v_{i+1}$ is a successor of $v_i$. The **depth** of a node $v$ is the maximum number of edges of a path from the root node to $v$.

**Definition 6** A **decorated graph** $\mathcal{G}$ is a graph in which every node has a decoration. For a node $v \in \mathcal{G}$, a **decoration** $\lambda(v)$ is a 5-tuple $([b_v, e_v], \rho(v), \mathcal{K}(v), \mathcal{L}(v), \mathcal{L}'(v))$, where $b_v(e_v)$ is a *constraint variable* denoting the beginning (ending) of the interval represented by the node $v$, $\rho(v)$ denotes the label of the node $v$ (where $\rho(v) \in \mathcal{E}$), $\mathcal{K}(v)$ denotes a formula associated with the node $v$, and $\mathcal{L}(v)$ and $\mathcal{L}'(v)$ denote a set of subformulas associated with the node $v$.

**Definition 7** A **temporal constraint** is a relation involving constraint variables which denote interval endpoints.

For example, the temporal constraint $b_v \geq b_u, e_v \leq e_u$ shows an interval relation between $[b_v, e_v]$ and $[b_u, e_u]$.

**Definition 8** A **tableau** for a given formula $\varphi$ is a tuple $\langle \mathcal{G}, \mathcal{C} \rangle$, where $\mathcal{G}$ denotes a decorated graph, and $\mathcal{C}$ denotes the set of temporal constraints in the graph $\mathcal{G}$.

### 4.2 Tableau Method

Let $\varphi$ be a formula to be checked for satisfiability over an interval $I_0$. The *initial tableau* for $\varphi$ is the tuple $\langle v_0, \mathcal{C}_0 \rangle$, where $v_0$ is the initial graph with the decoration $\lambda(v_0) = ([b_{v_0}, e_{v_0}], \rho(v_0), \mathcal{K}(v_0), \mathcal{L}(v_0), \mathcal{L}'(v_0))$ such that $\rho(v_0) = root, \mathcal{K}(v_0) = \varphi, \mathcal{L}(v_0) = \emptyset, \mathcal{L}'(v_0) = \emptyset$, and $\mathcal{C}_0$ is the initial set of temporal constraints such that $\mathcal{C}_0 = \{b_{v_0} = start(I_0), e_{v_0} = end(I_0)\}$. Assume $Q$ denotes the queue of nodes in $\mathcal{G}$ awaiting processing. Then, the initial value of $Q$ is $\{v_0\}$.

A tableau for $\varphi$ is a tuple $\langle \mathcal{G}, \mathcal{C} \rangle$, where $\mathcal{C}$ is obtained by expanding the initial constraint set $\mathcal{C}_0$ with temporal constraints in the existing nodes, and the decorated graph $\mathcal{G}$ is obtained by expanding the initial node $v_0$ through successive applications of the *expansion strategy* to existing nodes until no node remains to process. In other words, the expansion strategy is applied to every node in $Q$ until $Q = \emptyset$. When a node is selected, it is removed from $Q$.

During the application of the expansion strategy to a node, we need to solve the temporal constraints in $\mathcal{C}$. Remember that each node of the graph represents an interval. For our purposes, we model intervals as pairs of endpoints, which are distinct numbers on the real line. Let $T = \left\{b_{v_1}, \ldots, b_{v_n}, e_{v_1}, \ldots, e_{v_n}\right\}$ be a set of constraint variables. The constraints of a tableau can be represented as a Simple Temporal Problem (Dechter et al. 1991). If $n$ is the number of variables, then a solution to a STP (if there is any) can be found in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space. If the set of temporal constraints in $\mathcal{C}$ is inconsistent, then a solution will not be found, and we say $\mathcal{C}$ is not satisfiable.

In order to avoid infinite paths, and therefore to have a finite satisfying model we need to guarantee the termination of the proposed tableau method below. In the following we give a suitable *stoping condition* for the tableau procedure:

**Definition 9** A tableau $\langle \mathcal{G}, \mathcal{C} \rangle$ is **closed** if one of the following conditions hold:

- $\bot \in \mathcal{L}(v)$ for some node $v$ in $\mathcal{G}$,
- $\mathcal{C}$ is not satisfiable,
- The depth of the shortest path $v_0 \rightarrow \ldots \rightarrow v$ is more than $|\varphi|^2$ for some node $v$ in $\mathcal{G}$ (where $v_0$ is the root node.)

**Definition 10** A tableau is **open** if it is not closed .

Once the tableau procedure terminates, we check whether the tableau generated is open. For a given formula $\varphi$ if there is an open tableau, then $\varphi$ is satisfiable, and the satisfying model $\mathcal{M}$ is derived from the tableau. We do this by picking some solution $\sigma$, which assigns real values to constraint variables in $\mathcal{C}$. Let $J_v = [\sigma(b_v), \sigma(e_v)]$ be the interval represented by a node $v$ of $\mathcal{G}$. We construct a model $\mathcal{M}$ as follows: $\mathcal{M} = \{\langle J_v, \rho(v)\rangle \mid \text{for any } v \in \mathcal{G} \text{ s.t. } \rho(v) \notin \{root\}\}$. If the tableau is closed, then $\varphi$ is unsatisfiable.

*Expansion Strategy*

Let $\langle \mathcal{G}, \mathcal{C}\rangle$ be a tableau, $v$ be a node in $\mathcal{G}$ with $\lambda(v)$=$([b_v, e_v], \rho(v), \mathcal{K}(v), \mathcal{L}(v), \mathcal{L}'(v))$, and $Q$ be the queue of nodes awaiting processing. We say the **expansion strategy** for a node $v$ is defined as follows:

If the tableau is open, apply the following rules:

**Rule 1.** Set $Q := Q\backslash\{v\}$. If $\mathcal{L}(v)$ is empty, then apply the **interval relation rule** to the node $v$.

**Rule 2.** Let the Disjunctive Normal Form (DNF) of $\mathcal{K}(v)$ be $\psi_1 \vee ... \vee \psi_n$ where $\psi_i = \psi_{i1} \wedge \ldots \wedge \psi_{in_i} (n \geq 1, 1 \leq i \leq n \text{and} n_i \geq 1)$. Select some $i$, and set $\mathcal{L}'(v) := \left\{\psi_{i1}, \ldots, \psi_{in_i}\right\}$ and $\mathcal{L}(v) := \mathcal{L}(v) \cup \mathcal{L}'(v)$.[2]

**Rule 3.** Apply the **universal node expansion rule** to the node $v$.

**Rule 4.** Apply the **existential node expansion rule** to the node $v$.

In Rule 1 we check whether $\mathcal{L}(v)$ is empty. If $\mathcal{L}(v)$ is empty, then we know that the node $v$ has been newly created, and the interval relation rule has not been applied yet. By applying this rule we guess the interval relation between $v$ and any other node in $\mathcal{G}$. If $\mathcal{L}(v)$ is not empty, then we can conclude that the interval relation rule has been applied before. So we do not need to guess the interval relations again.

In Rule 2 we take the normal form of $\mathcal{K}(v)$ as disjunctions of subformulas (Each disjunct is composed of conjunctions of subformulas). According to this rule we non-deterministically select one of the disjuncts and assign it to $\mathcal{L}'(v)$, and add it to $\mathcal{L}(v)$. As can be seen, $\mathcal{L}'(v)$ only contains the selected disjunct. When the node is re-visited, we do not need to remember the previous value of $\mathcal{L}'(v)$. On the other hand, $\mathcal{L}(v)$ contains all subformulas assigned during the execution of the tableau procedure. Therefore, when the node is re-visited, we extend it with the new material in order to remember its previous value. It is also worth to mention that all of the elements of $\mathcal{L}(v)$ are expanded during the tableau construction.

*Interval Relation Rule*

The *interval relation rule* guesses the interval relation between the given node and all other nodes in the graph. Please note that we take Allen's interval relations as reference when considering an interval relation. Allen defines thirteen binary relations between intervals on a linear ordering, which are ‘before’, ‘after’, ‘meets’, ‘starts’, ‘during’, ‘finishes’, ‘equals’, ‘overlaps’, ‘met-by’, ‘started-by’, ‘finished-by’, ‘overlapped-by’ and ‘includes’.

[2] For simplicity we have not shown free variables in the formulas.

**Table 1** Allen's interval relations

| Basic interval relation | Relation symbol | Endpoint relations |
|---|---|---|
| $J$ before $J'$ | $R^b(J, J')$ | $e_J < b_{J'}$ |
| $J$ after $J'$ | $R^{\overline{b}}(J, J')$ | $e_{J'} < b_J$ |
| $J$ meets $J'$ | $R^m(J, J')$ | $e_J = b_{J'}$ |
| $J$ met-by $J'$ | $R^{\overline{m}}(J, J')$ | $e_{J'} = b_J$ |
| $J$ overlaps $J'$ | $R^o(J, J')$ | $b_J < b_{J'} < e_J < e_{J'}$ |
| $J$ overlapped-by $J'$ | $R^{\overline{o}}(J, J')$ | $b_{J'} < b_J < e_{J'} < e_J$ |
| $J$ during $J'$ | $R^d(J, J')$ | $b_J > b_{J'}, e_J < e_{J'}$ |
| $J$ includes $J'$ | $R^{\overline{d}}(J, J')$ | $b_{J'} > b_J, e_{J'} < e_J$ |
| $J$ starts $J'$ | $R^s(J, J')$ | $b_J = b_{J'}, e_J < e_{J'}$ |
| $J$ started-by $J'$ | $R^{\overline{s}}(J, J')$ | $b_{J'} = b_J, e_{J'} < e_J$ |
| $J$ finishes $J'$ | $R^f(J, J')$ | $b_J > b_{J'}, e_J = e_{J'}$ |
| $J$ finished-by $J'$ | $R^{\overline{f}}(J, J')$ | $b_{J'} > b_J, e_{J'} = e_J$ |
| $J$ equals $J'$ | $R^e(J, J')$ | $b_J = b_{J'}, e_J = e_{J'}$ |

Allen's approach to reasoning about time is based on the notion of time intervals and binary relations on them. Given two time intervals, their relative positions can be described by exactly one of the elements of the set $\mathcal{R}$ of thirteen basic interval relations, where each basic relation can be defined in terms of its endpoint relations. For example, giving that $J$ and $J'$ denote the intervals $[b_J, e_J]$ and $[b_{J'}, e_{J'}]$, respectively, with $b_J, e_J, b_{J'}, e_{J'} \epsilon \mathbb{R}$. An *atomic formula* of the form $R(J, J')$, where $R \epsilon \mathcal{R}$ is said to be satisfied by an interpretation iff the interpretation of the intervals satisfies the endpoint relations specified in Table 1.

Let $\langle \mathcal{G}, \mathcal{C} \rangle$ be a tableau, and $v$ be a node in $\mathcal{G}$ with $\lambda(v) = ([b_v, e_v], \rho(v), \mathcal{K}(v), \mathcal{L}(v), \mathcal{L}'(v))$. Assume $\tau'$ is the corresponding inverted operator of $\tau$ (where $\tau \in \{<, \leq, =, \geq, >\}$). The *interval relation rule* for a node $v$ is defined as follows:

***For any*** node $u$ (except $v$) in $\mathcal{G}$

Nondeterministically guess the interval relation between $u$ and $v$:

- $v$ *before* $u$ : Set $\mathcal{C} := \mathcal{C} \cup \{e_v < b_u\}$.
- $v$ *meets* $u$ : Set $\mathcal{C} := \mathcal{C} \cup \{e_v = b_u\}$.
- $v$ *non-strict-during* $u$ : Set $\mathcal{C} := \mathcal{C} \cup \{b_v \geq b_u, e_v \leq e_u\}$, and add an edge from $u$ to $v(u \rightarrow v)$. **if** $\rho(v) = e$ and $\forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x)) \in \mathcal{L}(u)$, **then** set either $i)\mathcal{C} := \mathcal{C} \cup \left\{(e_v - b_v)\tau' k\right\}$; or $ii)\mathcal{C} := \mathcal{C} \cup \{(e_v - b_v)\tau k\}$ and $\mathcal{K}(v) := \mathcal{K}(v) \wedge \psi(x)$.
- $voverlapsu$ : Set $\mathcal{C} := \mathcal{C} \cup \{b_v < b_u < e_v < e_u\}$, and add an edge from $u$ to $v(u \rightarrow v)$.

$v\ non - strict - during\ u$ is true if either $v$ "equals" $u$, $v$ "during" $u$, $v$ "starts" $u$ or $v$ "finishes" $u$ is true (Since we consider the case "non-strict-during", we do not need to consider these cases separately.) The cases where $v$ "after" $u$, $v$ "met-by" $u$, $v$ "includes" $u$, $v$ "started-by" $u$, $v$ "finished-by" $u$ and $v$ "overlapped-by" $u$ can be dealt with similarly.

We remark that once we have guessed the interval relation, we expand $\mathcal{C}$ with the corresponding endpoint relation. For example, if we have guessed $v$ "before" $u$, then we expand $\mathcal{C}$ with $\{e_v < b_u\}$. When we say, for example $v$ "before" $u$, we actually mean that this interval relation holds between the intervals $J_v$ and $J_u$ represented by the nodes $v$ and $u$, respectively. For simplicity, we will use this adaption.

In the interval relation rule, we consider the possibility that $\mathcal{L}(u)$ of an existing node $u$ includes a universal subformula which might update the decoration of the node $v$. Consider, for example, the above case where $v$ *non-strict-during* $u$. In this case $S(x, y)$ is true ($x$ any $y$ are instantiated by $J_v$ and $J_u$, respectively). If $\rho(v) = e$, than we can easily see that $e(x)$ is true. Furthermore, if $\forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x)) \in \mathcal{L}(u)$, then we may update $\mathcal{K}(v)$ depending on whether $\ell(x)\tau k$ is true. Here, we have two choices: either $\ell(x)\tau k$ is false, or $\ell(x)\tau k$ is true. If we choose the latter, then $\mathcal{K}(v)$ must be updated with $\psi(x)$ due to the implication by the universal formula.

*Universal Node Expansion Rule*

The *universal node expansion rule* expands all universal subformulas in $\mathcal{L}'(v)$. Let $\langle \mathcal{G}, \mathcal{C} \rangle$ be a tableau, and $v$ be a node in $\mathcal{G}$ with $\lambda(v)$=$([b_v, e_v], \rho(v), \mathcal{K}(v), \mathcal{L}(v), \mathcal{L}'(v))$. Assume $\tau'$ is the corresponding inverted operator of $\tau$ (where $\tau \in \{<, \leq, =, \geq, >\}$). The *universal node expansion rule* for a node $v$ is defined as follows:

***For every*** $\xi \in \mathcal{L}'(v)$

- **if** $\xi = \forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$, **then for every** node $w$ in $\mathcal{G}$ with $\rho(w) = e$ and $w$ *non-strict-during* $v$, set either *i)* $\mathcal{C} := \mathcal{C} \cup \left\{(e_w - b_w)\tau' k\right\}$; or *ii)* $\mathcal{C} := \mathcal{C} \cup \{(e_w - b_w)\tau k\}$, $\mathcal{K}(w) := \psi(x)$ and $Q := Q \cup \{w\}$.

Where $w$ "non-strict-during" $v$ is true if $b_w \geq b_v, e_w \leq e_v \in \mathcal{C}$. Note that above instead of $\psi(x)$ we have assigned $\mathcal{K}(w) \wedge \psi(x)$ to $\mathcal{K}(w)$. Because , we re-visit the node $w$, and therefore we do not want to use the previous material in $\mathcal{K}(w)$. For this reason, we have assigned $\psi(x)$ to $\mathcal{K}(w)$ in order to process only $\psi(x)$.

As result of applying the universal node expansion rule, some of the existing nodes might be re-visited, which means we re-execute the expansion strategy for these nodes. In this case, interval relations will not be guessed again; but their decoration might get updated.

*Existential Node Expansion Rule*

The *existential node expansion rule* expands all existential subformulas in $\mathcal{L}'(v)$. Let $\langle \mathcal{G}, \mathcal{C} \rangle$ be a tableau, and $v$ be a node in $\mathcal{G}$ with $\lambda(v)$=$([b_v, e_v], \rho(v), \mathcal{K}(v), \mathcal{L}(v), \mathcal{L}'(v))$. Assume $\tau'$ is the corresponding inverted operator of $\tau$ (where $\tau \in \{<, \leq, =, \geq, >\}$). The *existential node expansion rule* for a node $v$ is defined as follows:

***For every*** $\xi \in \mathcal{L}'(v)$

- **if** $\xi = \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$, **then** add an immediate successor $w$ with $\lambda(w) = ([b_w, e_w], \rho(w), \mathcal{K}(w), \mathcal{L}(w), \mathcal{L}'(w))$, where $\rho(w) = e, \mathcal{K}(w) =$

$\psi(x)$, $\mathcal{L}(w) = \emptyset$, $\mathcal{L}'(w) = \emptyset$, set $\mathcal{C} := \mathcal{C} \cup \{b_w \geq b_v, e_w \leq e_v, (e_w - b_w)\tau k\}$, add an edge from $v$ to $w$ ($v \to w$), and set $Q := Q \cup \{w\}$.

The existential node expansion rule creates a new node, and $\mathcal{L}(w)$ and $\mathcal{L}'(w)$ are initially set to $\emptyset$. In the next run, we apply the expansion strategy to this node, and $\mathcal{L}(w)$ and $\mathcal{L}'(w)$ get updated according to Rule 2.

### 4.3 Tableau Method at Work

In this section we apply the proposed decision procedure to a satisfiable formula. The decision procedure constructs a tableau $\langle \mathcal{G}, \mathcal{C} \rangle$ through successive applications of the expansion strategy to existing nodes until no node remains to process.

Below we will not show free variables in the formulas to preserve the simplicity. Assume,

$$\begin{aligned}
\psi' &= \exists x (e'(x) \wedge S(x, y) \wedge \ell(x) \geq 0) \\
\psi'' &= \exists x (e''(x) \wedge S(x, y) \wedge \ell(x) \geq 0) \\
\psi_1 &= \exists x (e(x) \wedge S(x, y) \wedge \ell(x) \geq 0 \to \psi'(x)) \\
\psi_2 &= \forall x (e(x) \wedge S(x, y) \wedge \ell(x) \geq 0 \to \psi''(x)).
\end{aligned}$$

Let $\varphi = \psi_1 \wedge \psi_2$ be a formula to be checked for satisfiability over an interval $I_0$. The initial tableau for $\varphi$ is the tuple $\langle v_0, \mathcal{C}_0 \rangle$, where $v_0$ is the initial graph with the decoration $\lambda(v_0) = ([b_{v_0}, e_{v_0}], root, \varphi, \emptyset, \emptyset)$, and $\mathcal{C}_0 = \{b_{v_0} = start(I_0), e_{v_0} = end(I_0)\}$. Also, the initial value of $Q$ is $\{v_0\}$. Now, we will show how the expansion strategy is applied to existing nodes:

$$\downarrow \text{Apply Rule 1 to } v_0$$

Rule 1 sets $Q := Q \backslash \{v_0\}$, and applies the interval relation rule to $v_0$. Since $v_0$ is the only node in the graph, the interval rule does not do anything.

$$\downarrow \text{Apply Rule 2 to } v_0$$

Rule 2 sets $\mathcal{L}'(v_0) = \mathcal{L}(v_0) = \{\psi_1, \psi_2\}$.

$$\downarrow \text{Apply Rule 3 to } v_0$$

The universal node expansion rule does not do anything.

$$\downarrow \text{Apply Rule 4 to } v_0$$

The existential node expansion rule expands $\psi_1$ by adding a new node $v_1$ (see Fig. 2) with $\lambda(v_1) = ([b_{v_1}, e_{v_1}], e, \psi', \emptyset, \emptyset)$, and sets $\mathcal{C} := \mathcal{C}_0 \cup \{b_{v_1} \geq b_{v_0}, e_{v_1} \leq e_{v_0}, (e_{v_1} - b_{v_1}) \geq 0\}$ and $Q := Q \cup \{v_0\}$.

$$\downarrow \text{Apply Rule 1 to } v_1$$

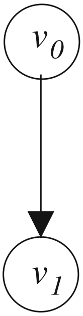

**Fig. 2** The graph after the existential node expansion rule expands $\psi_1$

Rule 1 sets $Q := Q\backslash\{v_1\}$, and applies the interval relation rule to $v_1$. The interval relation rule nondeterministically guesses the interval relation between the nodes $v_1$ and $v_0$. Assume the nondeterministic choice is $v_1$ "non-strict-during" $v_0$ (If the choice was either "before" , "meets" or "overlaps", then $\mathcal{C}$ would become inconsistent, and therefore the tableau would become closed.) In this case, $\mathcal{C}$ is set to $\mathcal{C} := \mathcal{C} \cup \{b_{v_1} \geq b_{v_0}, e_{v_1} \leq e_{v_0}\}$. Moreover, since $\rho(v_1) = e$ and $\forall x(e(x) \wedge S(x, y) \wedge \ell(x) \geq 0 \rightarrow \psi''(x)) \in \mathcal{L}(v_0)$, the interval rule sets $\mathcal{K}(v_1)$ to $\mathcal{K}(v_1) \wedge \psi''$ (i.e. $\mathcal{K}(v_1) := \psi' \wedge \psi''$), and $\mathcal{C}$ to $\mathcal{C} \cup (e_{v_1} - b_{v_1}) \geq 0$ (If the other choice was chosen, then $\mathcal{C}$ would be set to $\mathcal{C} \cup \{(e_{v_1} - b_{v_1}) < 0\}$. In this case, $\mathcal{C}$ would become inconsistent, and therefore the tableau would become closed.)

$\downarrow$ Apply Rule 2 to $v_1$

Rule 2 sets $\mathcal{L}'(v_1) = \mathcal{L}(v_1) = \{\psi', \psi''\}$.

$\downarrow$ Apply Rule 3 to $v_1$

Since there is no universal formula in $\mathcal{L}(v_1)$, the universal node expansion rule does not do anything.

$\downarrow$ Apply Rule 4 to $v_1$

The existential node expansion rule expands $\psi'$ and $\psi''$ by adding two new nodes $v_2$ with $\lambda(v_2) = ([b_{v_2}, e_{v_2}], e', \top, \emptyset, \emptyset)$ and $v_3$ with $\lambda(v_3) = ([b_{v_3}, e_{v_3}], e'', \top, \emptyset, \emptyset)$ (see Fig. 3), and sets $\mathcal{C} := \mathcal{C} \cup \{b_{v_2} \geq b_{v_1}, e_{v_2} \leq e_{v_1}, b_{v_3} \geq b_{v_1}, e_{v_3} \leq e_{v_1}, (e_{v_2} - b_{v_2}) \geq 0, (e_{v_3} - b_{v_3}) \geq 0\}$ and $Q := Q \cup \{v_2, v_3\}$.

$\downarrow$ Apply Rule 1 to $v_2$

Rule 1 sets $Q := Q\backslash\{v_2\}$, and applies the interval relation rule to $v_2$. Assume that the interval rule has chosen $v_2$ "non-strict-during" $v_0$ and $v_2$ "non-strict-during" $v_1$. In this case, $\mathcal{C}$ is set to $\mathcal{C} := \mathcal{C} \cup \{b_{v_2} \geq b_{v_0}, e_{v_2} \leq e_{v_0}, b_{v_2} \geq b_{v_1}, e_{v_2} \leq e_{v_1}\}$.

$\downarrow$ Apply Rule 2 to $v_2$

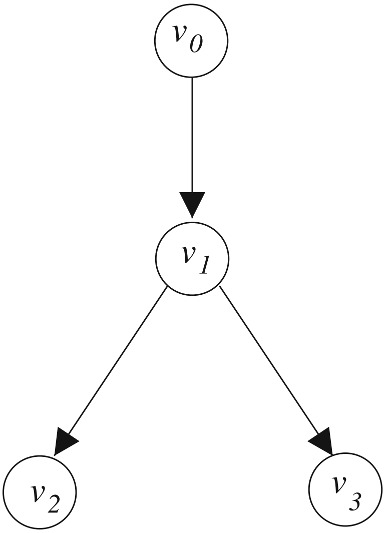

**Fig. 3** The graph after the existential node expansion rule expands $\psi'$ and $\psi''$

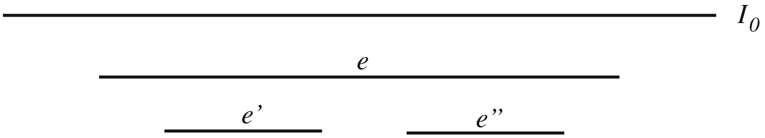

**Fig. 4** A model for $\varphi$

Rule 2 sets $\mathcal{L}'(v_2) = \mathcal{L}(v_2) = \{\top\}$.
The universal and existential node expansions rules do not do anything.

$\downarrow$ Apply Rule 1 to $v_3$

Rule 1 sets $Q := Q \backslash \{v_3\}$, and applies the interval relation rule to $v_3$. Assume that the interval rule has chosen $v_3$ "non-strict-during" $v_0$, $v_3$ "non-strict-during" $v_1$ and $v_2$ "before" $v_3$. In this case, $\mathcal{C}$ is set to $\mathcal{C} := \mathcal{C} \cup \{b_{v_3} \geq b_{v_0}, e_{v_3} \leq e_{v_0}, b_{v_3} \geq b_{v_1}, e_{v_3} \leq e_{v_1}, e_{v_2} \leq b_{v_3}\}$.

$\downarrow$ Apply Rule 2 to $v_3$

Rule 2 sets $\mathcal{L}'(v_3) = \mathcal{L}(v_3) = \{\top\}$ and $Q := Q \backslash \{v_3\}$.
Similarly, the universal and existential node expansions rules do not change anything. As can be seen, the tableau generated is open. Therefore, a satisfying model $\mathcal{M}$ can be derived from the tableau (Suppose we pick some solution for constraint variables in $\mathcal{C}$.) A model for the satisfiable formula $\varphi$ will look like Fig. 4.

## 4.4 Soundness and Completeness

The soundness and completeness of the proposed tableau method is proved below. But we first prove the termination of the method.

**Theorem 1** *The tableau method for EF terminates.*

*Proof* Let $\langle \mathcal{G}, \mathcal{C} \rangle$ be a tableau constructed by the tableau procedure for a given a formula $\varphi$. By the stoping condition in the tableau procedure every branch of the tableau is of finite length. We also know that every node of $\mathcal{G}$ has a finite outgoing degree. Therefore, the tableau method terminates. □

**Theorem 2** *Let $\varphi$ be an EF formula which has the form guaranteed by Lemma 1. $\varphi$ is satisfiable iff there is an open tableau for $\varphi$.*

*Proof (**Soundness** ($\Leftarrow$):)*

Suppose $\langle \mathcal{G}, \mathcal{C} \rangle$ is an open tableau for $\varphi$. We pick some solution $\sigma : \mathcal{V} \rightarrow \mathbb{R}$, which assigns real values to constraint variables in $\mathcal{C}$. Let $J_v = [\sigma(b_v), \sigma(e_v)]$ be the interval represented by the node $v$ of $\mathcal{G}$. We construct a model $\mathcal{M}$ as follows: $\mathcal{M} = \{\langle J_v, \rho(v) \rangle \mid \text{for any} v \in \mathcal{G} \text{s.t.} \rho(v) \notin \{root\}\}$.

Now we show that $\mathcal{M} \models \varphi[I_0]$ (where $I_0$ is the initial interval). We claim that for every $v$ in $\mathcal{G}, \mathcal{M} \models \mathcal{L}(v)[J_v]$. We show, by structural induction, that $\phi \in \mathcal{L}(v)$ implies $\mathcal{M} \models \phi[J_v]$. Note that, by construction of the tableau, $\mathcal{L}(v)$ comprises formulas of the forms $\top, \bot, \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$ and $\forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$.

***Base Case:***

$\phi = \top$ : Trivial

$\phi = \bot$ : Since $\langle \mathcal{G}, \mathcal{C} \rangle$ is an open tableau, by Definitions 9 and 10, $\bot \notin \mathcal{L}(v)$.

***Inductive Case:***

$\phi = \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$ : By the existential node expansion rule, there exists a node $w$ with $\rho(w) = e$ and $\mathcal{K}(w) = \psi(x)$. In addition, $\mathcal{C}$ contains $b_w \geq b_v, e_w \leq e_v$ and $(e_w - b_w)\tau k$. Let $\psi$ be $\psi_1 \vee \ldots \vee \psi_n$ where $\psi_i = \psi_{i1} \wedge \ldots \wedge \psi_{in_i}(n \geq 1, 1 \leq i \leq n$ and $n_i \geq 1)$. By Rule 2, $\psi_{i1}, .., \psi_{in_i} \in \mathcal{L}(w)$ for some $i(1 \leq i \leq n)$. By the inductive hypothesis, $\mathcal{M} \models \psi_{i1}[J_w] \wedge \ldots \wedge \mathcal{M} \models \psi_{in_i}[J_w]$. Therefore, $\mathcal{M} \models \psi[J_w]$. By construction, we have $\langle J_w, e \rangle \in \mathcal{M}$ with $|J_w| \, \tau k$ and $S(J_w, J_v)$. Thus, $\mathcal{M} \models \phi[J_v]$.

$\phi = \forall x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$ : By the construction of $\mathcal{M}$, for any $J \in \mathcal{I}$ if $\langle J, e \rangle \in \mathcal{M}$, then there exists a node $u$ in $\mathcal{G}$ such that $J_u = J$. According to the universal node expansion rule (or the interval relation rule) if $S(J_u, J_v)$, then we do either: $i$) set $\mathcal{C} := \mathcal{C} \cup \left\{(e_u - b_u)\tau' k\right\}$ ($\tau'$ is the corresponding inverted operator of $\tau$); or $ii$) set $\mathcal{C} := \mathcal{C} \cup \{(e_u - b_u)\tau k\}$ and $\mathcal{K}(u) := \psi(x)$ (We set $\mathcal{K}(u) := \mathcal{K}(u) \wedge \psi(x)$ when the interval relation rule applied.)

Assume $\ell(J_u)\tau k$ is false. Whatever the choice is, it is trivial to see that $\mathcal{M} \models e(J_u) \wedge S(J_u, J_v) \wedge \ell(J_u)\tau k \rightarrow \psi(J_u)$. Assume $\ell(J_u)\tau k$ is true. In this case, option $i$ mentioned above cannot have been selected. Otherwise, $\mathcal{C}$ would contain $\left\{(e_u - b_u)\tau' k\right\}$, and it would result in an inconsistency. So option $ii$ has been taken. In this case, we set $\mathcal{C} := \mathcal{C} \cup \{(e_u - b_u)\tau k\}$ and $\mathcal{K}(u) := \psi(x)(\mathcal{K}(u) := \mathcal{K}(u) \wedge \psi(x)$ in the case of the interval relation rule). Let $\psi$ be $\psi_1 \vee ... \vee \psi_n$ where $\psi_i = \psi_{i1} \wedge ... \wedge \psi_{in_i}(n \geq 1, 1 \leq i \leq n$ and $n_i \geq 1)$. By Rule 2, $\psi_{i1}, .., \psi_{in_i} \in \mathcal{L}(u)$ for some $i(1 \leq i \leq n)$. By the inductive hypothesis, $\mathcal{M} \models \psi(J_u)$. By construction, we have $\langle J_u, e\rangle \in \mathcal{M}$. We also know that $S(J_u, J_v)$ and $\ell(J_u)\tau k$. Therefore, for any witness $J_u, \mathcal{M} \models e(J_u) \wedge S(J_u, J_v) \wedge \ell(J_u)\tau k \rightarrow \psi(J_u)$. Thus, $\mathcal{M} \models \phi[J_v]$.

We have proved that for every $v$ in $\mathcal{G}, \mathcal{M} \models \mathcal{L}(v)[J_v]$. In particular, $\mathcal{M} \models \mathcal{L}(v_0)[I_0]$. We know that $\mathcal{K}(v_0) = \varphi$. Now assume $\varphi = \varphi_1 \vee ... \vee \varphi_n$, where $\varphi_i = \varphi_{i1} \wedge ... \wedge \varphi_{in_i}(n \geq 1, 1 \leq i \leq n\text{and}n_i \geq 1)$. According to Rule 2 $\mathcal{L}(v_0) = \{\varphi_{i1}, \ldots, \varphi_{in_i}\}$ for some value of $i$. Therefore, we can easily conclude that $\mathcal{M} \models \varphi[I_0]$.

**Completeness** ($\Rightarrow$) **:**

Suppose $\mathcal{M} \models \varphi[I_0]$. By Lemma 2 there exists a model $\mathcal{M}^* \subseteq \mathcal{M}$, with depth at most of order $|\varphi|^2$, such that $\mathcal{M}^* \models \varphi[I_0]$. We will show that there is an open tableau $\langle \mathcal{G}, \mathcal{C}\rangle$ for $\varphi$.

The *initial tableau* for $\varphi$ is the tuple $\langle v_0, \mathcal{C}_0\rangle$, where $v_0$ is the initial graph such that $\mathcal{K}(v_0) = \varphi$ and $\mathcal{L}(v_0) = \emptyset$, and $\mathcal{C}_0$ is the initial set of temporal constraints such that $\mathcal{C}_0 = \{b_{v_0} = start(I_0), e_{v_0} = end(I_0)\}$. A tableau $\langle \mathcal{G}, \mathcal{C}\rangle$ for $\varphi$ is obtained by expanding the initial node $v_0$ through successive applications of the expansion strategy to existing nodes until no node remains to process, and by expanding the initial constraint set $\mathcal{C}_0$ with temporal constraints in the existing nodes.

According to the expansion strategy we apply the interval relation rule to the node $v_0$ as $\mathcal{L}(v_0)$ is empty. But since there is only one node, $\mathcal{K}(v_0)$ does not get updated. Let the disjunctive normal form of $\mathcal{K}(v_0) = \varphi$ be $\varphi_1 \vee ... \vee \varphi_n$, where $\varphi_i = \varphi_{i1} \wedge ... \wedge \varphi_{in_i}(n \geq 1, 1 \leq i \leq n\text{and}n_i \geq 1)$. Since $\mathcal{M}^* \models \varphi[I_0]$, $\mathcal{M}^* \models \varphi_i[I_0]$ for at least one value of $i$. So in Rule 2 we pick this value of $i$, so that $\mathcal{L}(v_0) = \{\varphi_{i1}, ..., \varphi_{in_i}\}$.

Now, we claim that for each node $v$ in $\mathcal{G}$, there exists an interval $J_v$ such that $\mathcal{M}^* \models \mathcal{L}(v)[J_v]$ (Once we pick a witness $J_v$, it remains assigned to the node $v$ until the tableau procedure terminates.) We prove the claim by induction on the stage in tableau construction at which the node $v$ was created.

***Base case:***

Above we have shown that $\mathcal{M}^* \models \varphi_i[I_0]$ for some value of $i$, and $\mathcal{L}(v_0) = \{\varphi_{i1}, ..., \varphi_{in_i}\}$. So, it is trivial to see $\mathcal{M}^* \models \mathcal{L}(v_0)[I_0]$.

***Inductive case:***

Let $w$ be a node in $\mathcal{G}$ such that $\rho(w) = e$. Then $w$ must have been created by the existential node expansion rule applied to a node $v$ of which $w$ is a successor node. After the node $w$ has been created, we apply the expansion strategy to the node $w$. So we first apply the interval relation rule. Let us consider two cases:

$i)$ *Application of the interval relation rule adds no material to* $\mathcal{L}(w)$:[3] Assume $\mathcal{L}(w) = \{\psi_0\}$ where $\psi_0 = \psi_{01} \wedge ... \wedge \psi_{0n_0}(n_0 \geq 1)$. In this case, $\mathcal{L}(v)$ must contain $\xi = \exists x(e(x) \wedge S(x, y) \wedge \ell(x)\tau k \wedge \psi(x))$, where $\psi$ has the form

[3] Normally, if some material is added to $\mathcal{K}(w)$ as a result of applying the interval rule or the universal node expansion rule, $\mathcal{L}(w)$ does get updated in Rule 2 of the expansion strategy. Rule 2 selects some disjunct of

$\psi_0 \vee ... \vee \psi_l (l \geq 0)$. By the inductive hypothesis a witness $J_v$ is defined such that $\mathcal{M}^* \models \mathcal{L}(v)[J_v]$. Let $J_w$ be a witness for $x$. Thus, $\mathcal{M}^* \models \psi[J_w]$.

When the existential rule was applied to $v$, we set $\mathcal{K}(w) := \psi(x)$ and $\mathcal{C} := \mathcal{C} \cup \{b_w \geq b_v, e_w \leq e_v, (e_w - b_w)\tau k\}$. According to Rule 2 we select some of the disjunct of $\psi$, and extend $\mathcal{L}(w)$ with this disjunct. It is clear that $\psi_0$ is the subformula which was selected. So, $\mathcal{M}^* \models \psi_0[J_w]$. Hence, $\mathcal{M}^* \models \mathcal{L}(w)[J_w]$.

*ii*) *Application of the interval relation rule adds some material to* $\mathcal{L}(w)$: Assume $\mathcal{L}(w) = \{\psi_0, \psi_1, ..., \psi_m\}$ where $\psi_i = \psi_{i1} \wedge ... \wedge \psi_{im_i} (0 \leq i \leq m \text{and} m_i \geq 1)$, $\psi_0$ has been added to $\mathcal{L}(w)$ by applying the existential rule in $v$, and $\psi_1, ..., \psi_m$ have been added to $\mathcal{L}(w)$ by applying the interval relation rule to the node $w$. Above we have shown that $\mathcal{M}^* \models \psi_0[J_w]$.

According to the interval relation rule we guess the interval relation between $w$ and any node in $\mathcal{G}$. Assume for any $1 \leq j \leq m \psi_j$ has been added to $\mathcal{L}(w)$ as a result of guessing the interval relation between $w$ and a node $u_j$. Since $\mathcal{K}(w)$, and therefore $\mathcal{L}(w)$, has been updated, this relation must have been "non-strict-during". In this case, $\mathcal{L}(u_j)$ must contain $\xi = \forall x (e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$, where $\psi$ has the form $\psi_j \vee ... \vee \psi_{j+l} (l \geq 0)$. By the inductive hypothesis we have picked a witness $J_{u_j}$ such that $\mathcal{M}^* \models \mathcal{L}(u_j)[J_{u_j}]$; thus $\mathcal{M}^* \models \xi[J_{u_j}]$. We know that $S(J_w, J_{u_j})$ because in the interval rule we have guessed the relation between $J_w$ and $J_{u_j}$as "non-strict-during" (As we can see in the interval rule, $\mathcal{C}$ has been updated according to the corresponding non-deterministic choice of the relation.) We also know that $\ell(J_w)\tau k$ because we have selected the option *ii* in the interval relation rule, and set $\mathcal{C} := \mathcal{C} \cup \{(e_w - b_w)\tau k\}$ (Otherwise, $\mathcal{K}(w)$ could not have been updated). Therefore, $\mathcal{M}^* \models \psi[J_w]$.

When the interval rule was applied to $w$, we set $\mathcal{K}(w) := \mathcal{K}(w) \wedge \psi(x)$. It is clear that $\psi_j$ was selected when the Rule 2 the expansion strategy was applied. Thus, for any $1 \leq j \leq m$ $\mathcal{M}^* \models \psi_j[J_w]$. Hence, $\mathcal{M}^* \models \mathcal{L}(w)[J_w]$.

So, we have shown that once a node $w$ is created, and the expansion strategy is applied, it is true that $\mathcal{M}^* \models \mathcal{L}(w)[J_w]$. However, when new nodes are added to $\mathcal{G}$, $\mathcal{L}(w)$ might get updated through the application of the universal node expansion rule in these nodes. So, we must show that whenever new material is added to $\mathcal{L}(w)$, $\mathcal{M}^* \models \mathcal{L}(w)[J_w]$ remains true.

Now, assume $\mathcal{L}(w) = \{\psi_0, \ldots, \psi_m, \psi_{m+1}, \ldots, \psi_{m+n}\}$ where $\psi_i = \psi_{i1} \wedge \ldots \wedge \psi_{in_i} (0 \leq i \leq m + n \text{and} n_i \geq 1)$, and $\psi_{m+1}, \ldots, \psi_{m+n}$ have been added to $\mathcal{L}(w)$ by applying the universal node expansion rule to some nodes in $\mathcal{G}$. Above we have shown that $\mathcal{M}^* \models \{\psi_0, \ldots, \psi_m\}[J_w]$. Assume for any $m + 1 \leq k \leq m + n$, $\psi_k$ has been added to $\mathcal{L}(w)$ by applying the universal node expansion rule to a node $u_k$ in $\mathcal{G}$. In this case, $\mathcal{L}(u_k)$ must contain $\xi = \forall x (e(x) \wedge S(x, y) \wedge \ell(x)\tau k \rightarrow \psi(x))$, where $\psi$ has the form $\psi_k \vee ... \vee \psi_{k+l} (l \geq 0)$. By the inductive hypothesis we have picked a witness $J_{u_k}$ such that $\mathcal{M}^* \models \mathcal{L}(u_k)[J_{u_k}]$; thus $\mathcal{M}^* \models \xi[J_{u_k}]$. We know that $S(J_w, J_{u_k})$. We

Footnote 3 continued
this material, and updates $\mathcal{L}(w)$ with this disjunct. Here for our convenience we will say "a formula $\psi$ is added to $\mathcal{L}(w)$ by applying the interval or universal rule" simply to express the following process: "$\mathcal{K}(w)$ is updated as a result of applying the interval or universal rule. In Rule 2 $\psi$ is selected from the material added to $\mathcal{K}(w).\mathcal{L}(w)$ is updated with $\psi$."

also know that $\ell(J_w)\tau k$ because we have selected the option $ii$ of the universal rule, and set $\mathcal{C} := \mathcal{C} \cup \{(e_w - b_w)\tau k\}$ (Otherwise, $\mathcal{K}(w)$ could not have been updated.) Therefore, $\mathcal{M}^* \models \psi[J_w]$.

When the universal rule was applied to $u_k$, we set $\mathcal{K}(w) := \psi(x)$. It is clear that $\psi_k$ was selected when Rule 2 of the expansion strategy was applied. So, for any $m + 1 \leq k \leq m + n\mathcal{M}^* \models \psi_k[J_w]$. Hence, $\mathcal{M}^* \models \mathcal{L}(w)[J_w]$.

Therefore, we have proved that for each node $v$ in $\mathcal{G}$, there exists an interval $J_v$ such that $\mathcal{M}^* \models \mathcal{L}(v)[J_v]$.

Meanwhile, we know the depth of the model $\mathcal{M}^*$ is at most of order $|\varphi|^2$ by the assumption. Since for any node $v$ in $\mathcal{G}$ $\mathcal{M}^* \models \mathcal{L}(v)[J_v]$, $\perp$ cannot be contained in $\mathcal{L}(v)$. As we have a witness $J_v$ for each node $v$, we must have a solution for $\mathcal{C}$. Therefore, $\mathcal{C}$ must be satisfiable. Because none of the conditions in Definition 9 holds, it follows that $\langle \mathcal{G}, \mathcal{C} \rangle$ is an open tableau. □

### 4.5 Computational Complexity

**Theorem 3** *The satisfiability problem for EF is in NEXPTIME.*

*Proof* In Theorem 1 we show that the proposed method terminates. Now, we analyse its computational complexity. We now give a bound on the size of any tableau for $\varphi$.

In any node $v$ of $\mathcal{G}$ we convert $\mathcal{K}(v)$ into DNF, and in some cases conversion to DNF can lead to an exponential explosion of the formula. However, in the node expansion strategy we non-deterministically choose only one disjunct. Therefore, the out degree of any node is bounded by $|\varphi|$. We also know that the depth of the longest path in the tableau is bounded by $|\varphi|^2$ by Lemma 2. Thus, the size of the tableau is bounded by $|\varphi|^{|\varphi|^2} = 2^{|\varphi|^2 log_2 |\varphi|}$. So, the tableau procedure builds a tableau of size $2^{p(|\varphi|)}$ for some fixed polynomial $p$. We can say that if an EF formula $\varphi$ is satisfiable, then the tableau procedure construct a graph, from which a satisfying model $\mathcal{M}$ is extracted, of size bounded by $2^{p(|\varphi|)}$ for some fixed polynomial $p$. □

## 5 Conclusion and Future Work

In this paper we studied a new decidable fragment of first-order logic which is defined over interval structures. We called this new logic EF. EF is interpreted over a linear time flow with only finitely many events able to occur over a bounded-time interval. We showed that the depth of an EF model is polynomially bounded on the length of a given formula. We proved this by finding a reduced satisfying model, which has a polynomial depth bound on the size of the formula. This result played a key role in determining a limit on the size of a satisfying model.

We also proposed a terminating tableau system for EF, thus showing that its satisfiability problem is decidable. We, indeed, provided a complexity bound for satisfiability, showing that this problem can be solved in NEXPTIME. This is actually a common result for two variable fragments of first-order logic. This result shows that even a simple decidable fragment of first-order logic has NEXPTIME complexity.

We already know that fragments of first-order logic are closely related to modal logics; namely they extend modal logics. As a FOL fragment which was defined for specific purpose, the logic EF can be a general framework for event-based propositional interval logics, such as the ones defined in (Pratt-Hartmann 2005; Konur 2008). Such logics are decidable, and have a potential to be used in expressing the semantics of natural language constructions and specifying some properties of event-based real-time systems.

The results of this paper can be further developed in several directions. Some of the open problems are: finding a lower bound for the complexity, proving whether the satisfiability problem is NEXPTIME-complete, finding a finite axiomatisation for the logic EF, comparing expressive power of EF with other interval logics, extending EF with state types, notions of duration and accumulation and implementing the tableau method to have an automatic decision procedure.

**Acknowledgments** This work is partially funded by EPSRC under the Verifying Interoperability Requirements in Pervasive Systems (EP/F033567) project. The author would like to thank anonymous reviewers for their very helpful comments.

## References

Andreka, H., van Benthem, J., & Nemeti, I. (1996). Modal languges and Bounded Fragments of Predicate Logic. *Research Report ML-96-03, IILC.*

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence, 49*, 61–95.

Goranko, V., Montanari, A., & Sciavicco, G. (2003). Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science, 9*(9), 1137–1167.

Goranko, V., Montanari, A., Sciavicco, G., & Sala, P. (2006). A general Tableau method for propositional interval temporal logics: Theory and implementation. *Journal of Applied Logic, 4*(3), 305–330.

Grädel, E. (1999). On the restraining power of guards. *Journal of Symbolic Logic, 64*, 1719–1742.

Grädel, E., Kolaitis, P., & Vardi, M. (1997). On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic, 3*, 53–69.

Grädel, E., & Otto, M. (1999). On logics with two variables. *Theoretical Computer Science, 224*, 73–113.

Halpern, J.Y., & Shoham, Y. (1991). A propositional modal logic of time intervals. *Journal of the ACM, 38*(4), 935–962.

Konur, S. (2008). An interval logic for natural language semantics. *Advances in Modal Logic, 7*, 177–191.

Mortimer, M. (1975). On languages with two variables. *Zeitschr. F. Math. Logik U. Grundlagen d. Math., 21*, 135–140.

Moszkowski, B. (1983). *Reasoning about digital circuits*. Stanford University: PhD Thesis, Department of Computer Science.

Otto, M. (2001). Two variable first-order logic over ordered domains. *Journal of Symbolic Logic, 66*(2), 685–702.

Pratt-Hartmann, I. (2005). Temporal prepositions and their logic. *Artificial Intelligence, 166*(1–2), 1–36.

van Benthem, J. (1997). Dynamic Bits and Pieces. *Research Report, ILLC.*

Venema, Y. (1991). A modal logic for choppping intervals. *Journal of Logic and Computation, 1*, 453–476.