Halfway between Points and Intervals: A Temporal Logic Based on Ordered Trees

Ulle Endriss[1] and Dov Gabbay[2]

[1] Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2AZ (UK)
Email: ue@doc.ic.ac.uk
[2] Department of Computer Science, King's College London
Strand, London WC2R 2LS (UK)
Email: dg@dcs.kcl.ac.uk

Abstract. We present a new modal logic that is suitable to model complex systems evolving over time in a modular fashion. This logic may be regarded as the result of extending propositional linear temporal logic by a second dimension that allows us to "zoom" into states and thereby to further refine the specification of events associated with these states. In this sense, our logic may be described as an extended temporal logic that combines features from point-based and interval-based temporal logics. From an abstract point of view, our logic is best described as a modal logic over frames that are ordered trees.

1 Introduction

Temporal logics have been highly successful in several areas of computer science, in particular the specification and verification of reactive systems [6, 13]. For many applications the simplified view of time as a sequence of *points* turns out to be an adequate abstraction of reality. Notably in the area of temporal reasoning in artificial intelligence, however, authors have argued for systems based on time *intervals* rather than points [1]. Also in software engineering the difficulties associated with modelling the refinement of a system specification using a point-based temporal logic are widely recognised as an important problem [8]. Clearly, intervals provide us with a richer temporal representation formalism than simple point-based temporal logics.

On the downside, interval temporal logics are typically *undecidable* [11, 12, 16]. In this paper, we propose a middle way by introducing a temporal logic that is essentially point-based, but also integrates *some* of the desirable features of interval temporal logics, *without* giving up decidability. This logic may be regarded as the result of extending propositional linear temporal logic by a second dimension that allows us to "zoom" into states (which we may interpret as time intervals) and thereby to further refine the specification of events associated with these intervals (by explicitly speaking about their subintervals). As we shall see, another characterisation of this logic would be that of a *modal logic of ordered trees*. The purpose of this paper is to introduce the syntax and semantics of this new logic and to demonstrate its potential as a temporal representation language, mostly by example. For further details and proofs of the technical results (that we shall mention only in passing here) we refer to [7].

This paper is structured as follows. In Sec. 2, after a brief discussion of the respective merits and deficiencies of point-based and interval-based approaches to modelling time, we motivate the idea of adding a zoom to linear temporal logic and give an informal account of the resulting modal logic of ordered trees. This is complemented by a formal definition of the logic in Sec. 3. A number of issues relevant to using this logic as a temporal representation formalism are discussed in Sec. 4. We conclude in Sec. 5 with a brief review of related work.

2 From Points and Intervals to Ordered Trees

In this section we motivate our basic idea of developing a modal logic that is somewhere in between a simple point-based temporal logic on the one hand and a full interval logic on the other.

2.1 Points and Intervals

Given two distinct time *points* t_1 and t_2, the former may either lie *before* or *after* the latter. Furthermore, if time is assumed to be discrete, the two may also *meet*, i.e. one of them may lie immediately before the other.

In contrast, an *interval* may not only lie *before* another interval, but the two may also *meet* or *overlap*, or the former may either *start*, *finish*, or take place *during* the latter. In fact, there are exactly 13 different interval relations (the six aforementioned, their inverses, and equality). These relations are often referred to as *Allen relations*, due to Allen's influential paper [1]. An example for an interval logic is the modal logic of time intervals proposed by Halpern and Shoham [11], which is a multi-modal logic equipped with modal operators for each one of the 13 Allen relations (some of which are defined operators).

Both point-based and interval-based temporal logics (such as Halpern and Shoham's logic) allow us to model situations where one event takes place *before* (and possibly immediately before) another event. However, a logic where primitive units of time are intervals goes beyond simple point-based temporal logics in the following two crucial ways:

- we may express that two time units *overlap*; and
- we may express that a time unit can be *decomposed* into several others.

In addition to that, the full range of interval relations would also allow us to distinguish, for instance, between subintervals *starting* the dominating interval and subintervals that lie strictly *during* the dominating interval.

Halpern and Shoham's logic is strongly undecidable; so we cannot hope for a general modal interval logic with all these features that is computationally feasible. However, a *restricted* interval logic may still be of interest. Rather than using a logic that supports the full range of the 13 Allen relations, for certain applications it may be sufficient to consider a logic that allows us to speak at least about (i) *past* and *future* events (in the same way as a point-based logic would) and (ii) events taking place at *subintervals* of the reference interval—but a logic that cannot model the notion of overlapping intervals. We are now going to introduce such a logic as an extension of propositional linear temporal logic.

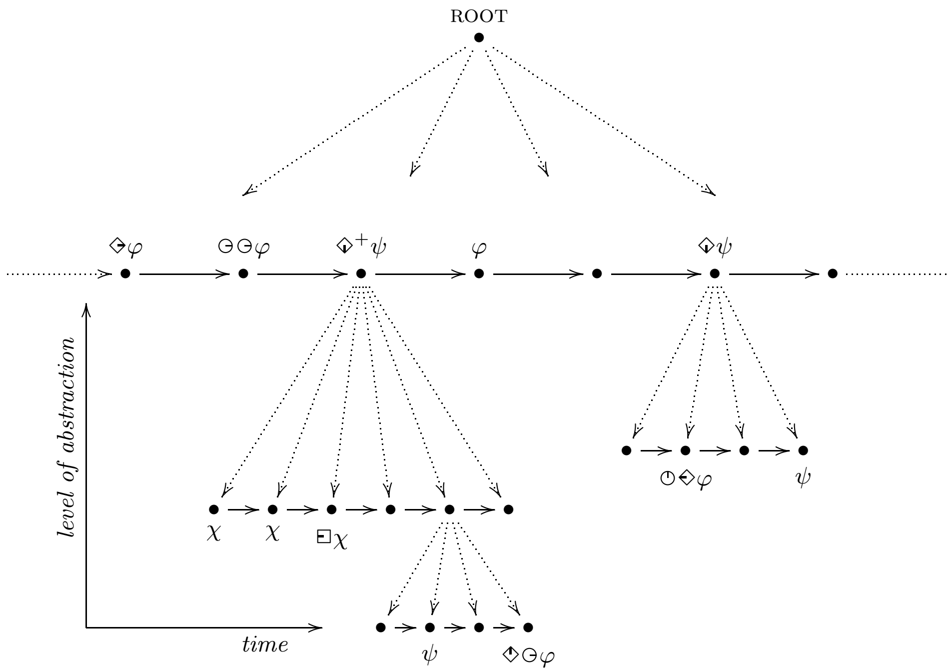

Fig. 1. Zooming in

2.2 Zooming in on Linear Temporal Logic

Despite their success in the area of systems specification, a drawback of standard point-based temporal logics is that they do not support the notion of refinement in a natural manner. There is no simple way to extend a given specification in, say, propositional linear temporal logic by the specification of a new subsystem. To overcome this problem, we propose to *add a zoom to linear temporal logic* by explicitly relating the state to be refined to another time line which represents the course of events taking place *during* that "state" (or rather the time interval associated with that state), at the next lower level of abstraction. The new states may themselves be refined, i.e. we may also zoom into the states on the second level of abstraction, and so forth. This idea is illustrated in Fig. 1.

The picture also shows some of the modal operators we propose to include into our new temporal language. The *horizontal* operators are those of linear temporal logic. For instance, we write $\Diamond_{\rightarrow}\varphi$ to say that φ is true at *some future* state (at the same level of abstraction), while $\ominus\varphi$ expresses that φ is true at the *next* state (again at the same level of abstraction).[3] There are similar operators available to speak about the past. As for the *vertical* operators, we write $\Diamond_{\downarrow}\varphi$ to say that φ is true at *some* state belonging to the *next lower* level of abstraction, and $\Diamond_{\downarrow}^{+}\varphi$ if φ is true at *some* state of *some lower* level of abstraction. To be able to move back up again in the hierarchy

[3] Note that we do not consider the *until*-operator here.

of states, we may use either $\overline{\bigcirc}$ (to refer to the *next higher* level of abstraction) or $\overline{\Diamond}$ (to refer to *some higher* level).

Linear temporal logics are based on linear orders. What is the structure underlying our extended temporal logic? If we add a “first level of abstraction” consisting of only a single state (which may be thought of as representing the specified system as a whole) to the kind of model we have described before, we end up with a tree-like structure where the children of each node are ordered. Hence, from an abstract point of view, our extended temporal logic may be characterised as a *modal logic of ordered trees*.

3 Syntax and Semantics of OTL

In this section, we introduce OTL, the modal logic of ordered trees, formally and (mostly) independently from its potential interpretation as an extended temporal logic based on decomposable time units. We assume some familiarity with elementary modal logic (see e.g. Blackburn *et al.* [3]).

3.1 Syntax

The language of OTL is built around a countable set of propositional letters. The set of *well-formed formulas* A of our logic is formed according to the following BNF production rule (where P stands for any propositional letter):

$$A \ ::= \ P \mid \neg A \mid A \wedge A \mid \overleftarrow{\bigcirc} A \mid \overrightarrow{\bigcirc} A \mid \overline{\bigcirc} A \mid \overleftarrow{\Diamond} A \mid \overrightarrow{\Diamond} A \mid \overline{\Diamond} A \mid \underline{\Diamond} A \mid \underline{\Diamond}^{+} A$$

Further propositional connectives (as well as $\bot$ and $\top$) are defined as syntactical abbreviations in the usual manner. Also, for each of the diamond-operators in our language we introduce a corresponding box-operator, e.g. $\overrightarrow{\Box}\,\varphi = \neg \overrightarrow{\Diamond} \neg\varphi$.

3.2 Ordered Trees

An ordered tree is a tree where the children of each node are ordered. The following definition of a tree is a slight modification of that given by Goldblatt [10]:[4]

Definition 1 (Trees). *A tree is a pair $\mathcal{T} = (T, R)$, where T is a set and R is an irreflexive binary relation over T satisfying the following conditions:*

1. *For every $t \in T$ there exists at most one $t' \in T$ with $(t', t) \in R$.*
2. *There exists a unique $r \in T$ such that $\{t \in T \mid (r, t) \in R^*\} = T$.*

The elements of T are called nodes. The element r from condition (2) is called the root of $\mathcal{T}$. R is called the child relation and also gives rise to the following: the parent relation R^{-1}, the descendant relation R^+, the ancestor relation $(R^{-1})^+$, and the sibling relation $R^{-1} \circ R$.

[4] Here and in the sequel we refer to a number of relational algebra operations: the inverse of a relation R is denoted by R^{-1}; the composition of two relations R_1 and R_2 is written as $R_1 \circ R_2$; R^* denotes the reflexive transitive closure and R^+ the non-reflexive transitive closure of R, respectively.

So by referring, for instance, to a node t_1 as the parent of t_2, we mean that $(t_1, t_2) \in R$. Node t_2 is then called a child of t_1. Similarly, if $(t_1, t_2) \in R^+$ then t_1 is called an ancestor of t_2, and t_2 is called a descendant of t_1. Two nodes that share the same parent are called siblings. Starting at any node, if we first move up one level using the parent relation and then one level down via the child relation, we will end up at a sibling of the first node; that is, if we have $(t_1, t_2) \in R^{-1} \circ R$ then t_1 is a sibling of t_2 and vice versa. The nodes in a tree may be considered as being "generated" by the root node via the transitive closure of the child relation R. This excludes certain kinds of structures, such as trees (in a wider sense of the word) where the child relation could, for instance, be dense. In other words, we only consider trees of order-type $\mathbb{N}$.

We are now ready to define ordered trees as trees with an additional (linear) order over the children of each node. Observe that the sibling relation $R^{-1} \circ R$ is an equivalence relation over the set of nodes excluding the root. For a given node t the "quasi-equivalence class" $[t]_{R^{-1} \circ R} = \{t' \in T \mid (t, t') \in R^{-1} \circ R\}$ is the set of siblings of t (including t itself); only if t is the root of the tree then $[t]_{R^{-1} \circ R}$ is the empty set.

Definition 2 (Ordered trees). *An ordered tree is a triple $\mathcal{T} = (T, R, S)$ where (T, R) is a tree, $S \subseteq R^{-1} \circ R$, and $([t]_{R^{-1} \circ R}, S)$ is a strict linear order for every $t \in T$. If $(t_1, t_2) \in S$ then t_1 is called a left sibling of t_2, and t_2 is called a right sibling of t_1. If furthermore $(t_1, t_2) \notin S \circ S$ then t_1 is called the left neighbour of t_2, and t_2 is called the right neighbour of t_1.*

According to this definition the horizontal relation declared over the children of a given node could be *any* strict linear order. In particular, it is not required to be discrete. This means, for example, that it is possible that a node has righthand siblings but no righthand neighbour, namely when that node is dense, i.e. when there are infinitely many nodes between itself and any of its righthand siblings. That is, OTL will turn out to be an extension of linear temporal logic over *general* flows of time.

3.3 Semantics

We can now move on to the definition of models and the notion of truth of a formula in a model. OTL is a standard modal logic and frames in this modal logic are ordered trees.

Definition 3 (Models). *An ordered tree model is a pair $\mathcal{M} = (\mathcal{T}, V)$, where $\mathcal{T}$ is an ordered tree and V is a valuation, that is, a mapping from propositional letters to subsets of the set of nodes in $\mathcal{T}$.*

Definition 4 (Truth). *We inductively define the notion of truth of a formula in a model $\mathcal{M} = (\mathcal{T}, V)$ with $\mathcal{T} = (T, R, S)$ at a node $t \in T$ as follows:*

1. *$\mathcal{M}, t \models P$ iff $t \in V(P)$ for propositional letters P;*
2. *$\mathcal{M}, t \models \neg\varphi$ iff $\mathcal{M}, t \not\models \varphi$;*
3. *$\mathcal{M}, t \models \varphi \wedge \psi$ iff $\mathcal{M}, t \models \varphi$ and $\mathcal{M}, t \models \psi$;*
4. *$\mathcal{M}, t \models \bigcirc\!\!\!\!\uparrow\,\varphi$ iff t is not the root of $\mathcal{T}$ and $\mathcal{M}, t' \models \varphi$ for the parent t' of t;*
5. *$\mathcal{M}, t \models \Diamond\!\!\!\uparrow\,\varphi$ iff t has got an ancestor t' with $\mathcal{M}, t' \models \varphi$;*
6. *$\mathcal{M}, t \models \bigcirc\!\!\!\!\leftarrow\varphi$ iff t has got a left neighbour t' with $\mathcal{M}, t' \models \varphi$;*
7. *$\mathcal{M}, t \models \Diamond\!\!\!\leftarrow\varphi$ iff t has got a left sibling t' with $\mathcal{M}, t' \models \varphi$;*

8. $\mathcal{M}, t \models \ominus\varphi$ *iff* t *has got a right neighbour* t' *with* $\mathcal{M}, t' \models \varphi$;
9. $\mathcal{M}, t \models \Diamond_{\rightarrow}\varphi$ *iff* t *has got a right sibling* t' *with* $\mathcal{M}, t' \models \varphi$;
10. $\mathcal{M}, t \models \Diamond_{\downarrow}\varphi$ *iff* t *has got a child* t' *with* $\mathcal{M}, t' \models \varphi$;
11. $\mathcal{M}, t \models \Diamond_{\downarrow}^{+}\varphi$ *iff* t *has got a descendant* t' *with* $\mathcal{M}, t' \models \varphi$.

It follows, for instance, that a formula of the form $\Box_{\rightarrow}\varphi$ is true at a node t in a given model $\mathcal{M}$ iff φ is true at every righthand sibling of t. The notion of *satisfiability* and *validity* are defined as usual [3]. Furthermore, a formula φ is called *globally true* in a given model $\mathcal{M}$ iff φ is true at every node in $\mathcal{M}$. A formula φ is called *valid in an ordered tree* $\mathcal{T}$ iff φ is true at every node in every model based on $\mathcal{T}$.

In [7] a complete axiomatisation of a large sublogic of the logic presented here (namely the fragment of OTL excluding the descendant operator $\Diamond_{\downarrow}^{+}$) is given. We hope to address the issue of extending this axiomatisation to the full logic in our future work.

3.4 Decidability

We note here that it has been shown in [7] that OTL is a *decidable* logic. The proof amounts, essentially, to a reduction to Rabin's Theorem on the decidability of S2S, the monadic second-order theory of two successor functions [17]. This reduction builds on the well-known construction used to prove the decidability of propositional linear temporal logic over the rationals [9, 17]. An important intermediate lemma establishes that every formula with any ordered tree model at all will also have a model based on an ordered tree with (at most) a countable number of nodes. This allows us to lift the decidability result obtained using reduction (which applies only to OTL over countable trees) to the general case. Using similar arguments, we can show that OTL restricted to certain classes of ordered trees (such as those where sequences of siblings are required to be discrete orders) is also decidable [7].

If we compare our logic with the undecidable interval logic of Halpern and Shoham [11], we observe two important differences that may account for the different computational behaviour of the two logics. Firstly, OTL cannot model the concept of overlapping time intervals. Secondly, if we consider the set of intervals represented by some ordered tree model, that set does not (necessarily) contain all the intervals we could construct from the endpoints of the intervals already in the set. For instance, just because a model does feature two neighbouring intervals does not mean that that model will also include the interval we would obtain by conjoining those two intervals (namely if the two nodes in question have further siblings in the tree). This is different for Halpern and Shoham's logic, which—intuitively—makes the latter the more complex system.

Such intuitive explanations aside, a possible objection to a decidability proof by reduction to a very powerful system like S2S would be that it provides only little insight as to *why* the logic in question is decidable. We are currently investigating options for proving decidability of OTL directly, namely by using techniques inspired by the work of Sistla and Clarke [19] on upper complexity bounds for linear temporal logics. Such a proof may not only help to understand OTL better, but it may also provide a good starting point for analysing the computational complexity of our logic.

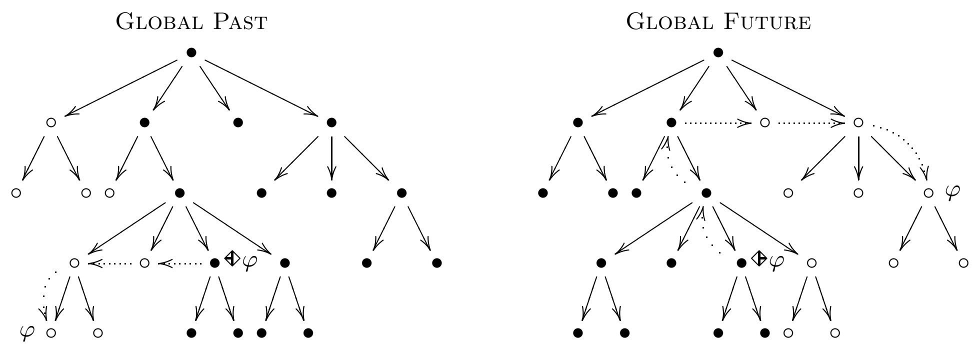

Fig. 2. Scope of the global past and future modalities

4 OTL as a Temporal Representation Language

Our inspiration for devising a modal logic of ordered trees was to extend propositional linear temporal logic in a way that allows us to describe systems evolving over time in a modular fashion. Under this view of OTL, *nodes* in a tree represent *time intervals*. In this section, we discuss some of the issues relevant to interpreting OTL as a *temporal* logic.

If t_1 is an ancestor of t_2 then we can think of the interval corresponding to t_2 as taking place *during* that corresponding to t_1. If t_1 is a lefthand sibling of t_2 then the interval corresponding to t_1 takes place *before* that corresponding to t_2. If t_1 and t_2 are neighbours then the corresponding intervals *meet*, that is, t_1 takes place immediately before t_2. As we shall see next, we can apply the notion of one event taking place before another also to nodes that are not siblings.

4.1 Global Past and Future Modalities

We can combine basic OTL modalities in such a way as to catch not only the siblings to, say, the right of a given node t, but all the nodes that lie somewhere to the right of t anywhere in the tree. In the context of our temporal interpretation of OTL, such an operator may be described as a "global future" modality. We think of the nodes representing time points in the future of t as those nodes we can reach by first going up an unspecified number of levels in the tree (or staying at t), then making at least one step to the right (that is, moving to a righthand sibling), and finally moving down any number of levels (or possibly staying where we are). Analogously, we can define a global past modality to refer to nodes anywhere to the left of the current node.

Let $\Diamond_{\uparrow}^* \varphi = \varphi \vee \Diamond_{\uparrow} \varphi$ and $\Diamond_{\downarrow}^* \varphi = \varphi \vee \Diamond_{\downarrow}^+ \varphi$ for all formulas φ. We are now in a position to give syntactical definitions of an existential global past operators $\Diamond_{\Leftarrow}$ and a corresponding global future operator $\Diamond_{\Rightarrow}$ as follows:

$$\Diamond_{\Leftarrow} \varphi \;=\; \Diamond_{\uparrow}^* \Diamond_{\leftarrow} \Diamond_{\downarrow}^* \varphi \qquad\qquad \Diamond_{\Rightarrow} \varphi \;=\; \Diamond_{\uparrow}^* \Diamond_{\rightarrow} \Diamond_{\downarrow}^* \varphi$$

Box-operators dual to these may be defined in the usual manner. Fig. 2 provides a graphical depiction of the scope of our global past and future operators. Take, for instance, the tree on the righthand side. The formula $\Diamond\!\!\!\!\vdash\varphi$ is supposed to hold at one of the nodes in the tree. To get to the witness formula φ shown in the picture, we first have to move up two levels in the tree, then move two steps to the right, and finally move down again by one level. The nodes in the scope of $\Diamond\!\!\!\!\vdash$ are shown as little empty circles ($\circ$), while the rest of the nodes are represented as filled circles ($\bullet$). The lefthand side of the picture shows a similar example for the case of the existential global past modality.

4.2 Ontological Considerations

Suppose φ is a formula that is true at some node t. If we think of the children of t as a more fine-grained representation of the time interval associated with t, then should this not imply that φ must also hold at any of the children of t? If φ stands for *"Mary travelled all the way from London to Vienna"* then it may very well be true at some node t, but false at one of t's children (that is, during a time period within the one corresponding to t). In fact, we might even want to enforce $\neg\varphi$ for all descendants of t as Mary cannot complete her entire journey both over the duration of t and also in a shorter period within t. If, on the other hand, φ stands for a proposition like *"Mary has long blonde hair"* then we *do* want φ to hold at descendants of t as well. In the literature on foundations of planning, propositions of the latter kind have been called *properties* or *facts*, while propositions like *"Mary travelled all the way from London to Vienna"* are known as *events* [2, 14, 18].

Such ontological distinctions often form an integral part of the definition of a planning formalism. In Allen's work [2], for example, the definition of the predicates HOLDS and OCCUR, which are used to distinguish between different types of propositions, is central to the entire system. HOLDS is applied only to properties and OCCUR is reserved for the use with occurrences (an ontological category that includes events). This approach has been criticised by Shoham [18], who argues that, in the first instance, one should rather not make any specific commitments to ontology, but then allow for the definition of a fine ontological structure on top of the basic formalism. In the sequel, we demonstrate how this may be done in the case of OTL.

A *property* like *"Mary has long blonde hair"* is *homogeneous:* a homogeneous proposition φ is true iff it is true at every time interval within the reference interval. Following Shoham [18] we define homogeneous propositions as propositions that are both *downward-hereditary* and *upward-hereditary.* A proposition is called downward-hereditary if whenever it holds at some node it also holds at all descendants of that node. Analogously, a proposition is called upward-hereditary if whenever it holds at all descendants of a given node it also holds at that node itself (assuming that node has any descendants at all).

$$\begin{aligned} \text{DOWN-HERED}(\varphi) &= \varphi \rightarrow \boxdot^{+}\varphi \\ \text{UP-HERED}(\varphi) &= \Diamond\top \rightarrow (\boxdot^{+}\varphi \rightarrow \varphi) \\ \text{HOM}(\varphi) &= \text{DOWN-HERED}(\varphi) \wedge \text{UP-HERED}(\varphi) \end{aligned}$$

That is, φ is a homogeneous proposition (with respect to a given model) iff $\text{HOM}(\varphi)$ is globally true in that model.

An *event* such as *"Mary travelled all the way from London to Vienna"* should not hold over two time intervals one of which properly contains the other. Shoham [18] calls propositions of this type *gestalt*.[5] In the context of OTL we can interpret this condition as follows. A proposition φ is gestalt iff whenever φ is true at a node t then it must be false at all ancestors and descendants of t:

$$\textsc{gestalt}(\varphi) \;=\; \varphi \;\rightarrow\; (\boxed{\uparrow}\neg\varphi \wedge \boxed{\downarrow}^{+}\neg\varphi)$$

Another ontological concept that can be expressed is locality. A proposition φ is called *local* iff φ is true during a particular interval whenever it is true at the beginning of that interval [11]. In our system, the beginning of an interval may be identified with the leftmost child of a node. Observe that a node is a leftmost child iff it satisfies the formula $\boxed{\leftarrow}\bot$. For better readability, we abbreviate $\textsc{leftmost} = \boxed{\leftarrow}\bot$. The following definition takes into account that a node does not necessarily have to have a leftmost child:

$$\textsc{local}(\varphi) \;=\; (\langle\downarrow\rangle\textsc{leftmost} \wedge \varphi) \;\leftrightarrow\; \langle\downarrow\rangle(\textsc{leftmost} \wedge \varphi)$$

What we have seen here are just examples. Hopefully, they give some impression of the variety of options available to us when using OTL as a temporal representation formalism.

5 Conclusion

We have proposed a new modal logic that, in our view, provides an interesting alternative to some of the well-studied interval temporal logics. Primitive time units of this logic are somewhere between points and intervals: they may be decomposed (like intervals, unlike points), but they cannot overlap (like points, unlike intervals). The ability to decompose basic time units allows for the modelling of systems in a *modular* fashion, in a way that is not possible for a logic with a strictly point-based semantics. On the other hand, by restricting the expressive power (in particular, by excluding the notion of overlapping time periods), we have been able to retain decidability of our logic, a crucial feature not shared by many other interval temporal logics. We conclude by reviewing some related systems.

Moszkowski's ITL [16], in particular if extended with a *projection operator*, is an interval logic suitable to reason about the decomposition of time units into smaller parts. However, unlike our logic, ITL does not provide means of referring to intervals outside the reference interval, i.e. it is not possible to reason about either future or past events. Another logic that allows for the decomposition of time periods is the layered temporal logic of time granularities proposed by Montanari [15]. This logic is particularly suited for domains where reasoning about *fixed* time units and their relation to each other is required (e.g. days, hours, and minutes).

The logic that seems to come closest to our logic (that we are aware of) is probably the logic of finite trees of Blackburn, Meyer-Viol and de Rijke [5]. Important differences include that the horizontal dimension in our logic need not be discrete and that branches may be of infinite length. In fact, such restrictions to the class of admissible frames can be expressed within OTL. For instance, the formula $A \rightarrow \langle\downarrow\rangle^{*}(A \wedge \boxed{\downarrow}^{+}\neg A)$

[5] Related to this, Shoham also defines the class of *solid* propositions, which are propositions that cannot be true at overlapping intervals. This would provide an alternative way to characterise events, but overlapping intervals cannot be represented in OTL.

will be valid in a tree whenever that tree is a tree of finite depth [4, 7]. The formula essentially expresses that for every node satisfying A there must be a "final" descendant along any branch that also satisfies A. Discreteness is characterised by the formula $⊟(⊟A \rightarrow A) \rightarrow (◈⊟A \rightarrow ⊟A)$, that is, by the usual axiom schema for discrete flows of time [10]. Apart from such technical differences, we consider our interpretation of this kind of logic as a *temporal* logic (with time running orthogonally to the tree structure) an important conceptual contribution.

Acknowledgements. The work reported here has been partially supported by UK EPSRC grants GR/R45369/01 and GR/N23028/01.

References

1. J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. J. F. Allen. Towards a General Theory of Action and Time. *Artificial Intelligence*, 23:123–154, 1984.
3. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
4. P. Blackburn and W. Meyer-Viol. Linguistics, Logic and Finite Trees. *Logic Journal of the IGPL*, 2:3–29, 1994.
5. P. Blackburn, W. Meyer-Viol, and M. de Rijke. A Proof System for Finite Trees. In H. Kleine Büning, editor, *Computer Science Logic*, volume 1092 of *LNCS*, pages 86–105. Springer-Verlag, 1996.
6. E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
7. U. Endriss. *Modal Logics of Ordered Trees*. PhD thesis, King's College London, Department of Computer Science, January 2003.
8. J. L. Fiadeiro and T. Maibaum. Sometimes "tomorrow" is "sometime": Action Refinement in a Temporal Logic of Objects. In D. Gabbay and H. J. Ohlbach, editors, *Temporal Logic*, volume 827 of *LNAI*, pages 48–66. Springer-Verlag, 1994.
9. D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 1. Oxford University Press, 1994.
10. R. Goldblatt. *Logics of Time and Computation*. CSLI, 2nd edition, 1992.
11. J. Y. Halpern and Y. Shoham. A Propositional Modal Logic of Time Intervals. *Journal of the ACM*, 38(4):935–962, 1991.
12. K. Lodaya. Sharpening the Undecidability of Interval Temporal Logic. In J. He and M. Sato, editors, *Proceedings of the 6th Asian Computing Science Conference (ASIAN-2000)*, volume 1961 of *LNCS*, pages 290–298. Springer-Verlag, 2000.
13. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.
14. D. V. McDermott. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101–155, 1982.
15. A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 1996.
16. B. Moszkowski. A Temporal Logic for Multilevel Reasoning about Hardware. *IEEE Computer*, 18(2):10–19, 1985.
17. M. O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the AMS*, 141:1–35, 1969.
18. Y. Shoham. Temporal Logics in AI: Semantical and Ontological Considerations. *Artificial Intelligence*, 33:89–104, 1987.
19. A. P. Sistla and E. M. Clarke. The Complexity of Propositional Linear Temporal Logics. *Journal of the ACM*, 32(3):733–749, 1985.