A Survey on Temporal Logics

Savas Konur

Department of Computer Science
University of Liverpool
October 2008

Contents

Abstract

This paper surveys main and recent developments on temporal logics in a broad sense by presenting various formal systems dealing with various time structures, and discussing important features, such as (un)decidability results, expressiveness and axiomatization systems.

1 Introduction

The study of time spans a variety of different disciples, such as physics, philosophy, literature, computer science, etc. Time has been one of the most paradoxical concepts of philosophy throughout history. A satisfactory definition of time has not been presented since ancient times. Each definition has covered some aspects of time, while excluding others. Plato defined it as the 'moving image of eternity', and Aristo described it as 'the number of motion with respect to earlier and later' (see [Whi72]). In order to establish a common language for time, the concept of time has been studied in various disciplines.

Temporal logic focuses on propositions whose truth values depend on time. Temporal propositions typically contain some reference to time conditions, while classical logic deals with timeless propositions. As classical logic formulas can characterize static states and properties, temporal logic formulas can describe sequences of state changes and properties of behaviors, and, hence, can span a wide range of problems in various fields with a richer notation.

A temporal logic basically results from an extension of a classical propositional or predicate logic with temporal quantifiers introducing temporal modalities. Due to its temporal quantifiers temporal logic is a convenient and appropriate means to reason with time-related propositions. Indeed, classical logic can also handle temporal properties, but the formulas tend to be complicated since points of time have to be explicitly represented in the underlying universe.

Temporal logic has been an important subject of philosophy. Even some of the ancient philosophizers used some form of temporal logic. During the middle ages logicians resumed and refined the ancient approaches. In modern times, the interest in symbolic logic grew during the first half of the 20th century, and after some delay new modal and temporal logic approaches occurred. First publications date back to the 1940's.

Although various aspects of time and logic have been studied by many scientists, an up-to date comprehensive analysis of logic of time does not exist in the literature. Some important historical surveys, like [OH95], have been presented; but these do not include recent developments in the field. Several formal approaches can be found in the literature ([CH04, GMS04]) but these mainly concentrate on specific formal systems over specific structures of time; therefore, they do not contain a broad analysis. The aim of this paper is to outline main and recent developments in the field in a broad sense by presenting various formal systems dealing with various time structures, and discussing important features, such as (un)decidability results, expressiveness and axiomatization systems.

The paper is organized as follows: In Section 2 we introduce various ontological issues which are the common criteria to distinguish among logic systems. In Section 3, 4, 5 and 6 we discuss propositional temporal logics, first order temporal logics, branching time and partial order temporal logics, and interval temporal logics, respectively. Section 7 contains some concluding remarks.

2 Temporal Ontologies

Temporal logic systems can be classified along various dimensions: linear versus branching; propositional versus first-order; finite versus infinite; discrete versus continuous, etc. The most common criteria distinguishing temporal logics are given below (see [Lam80, Mos83, Pnu85, LPZ85, HS86, BKP86, HS91, Eme95, Ven98, LP00]):

2.1 Choice between Branching Time and Linear Time

In a temporal logic the structure of time can be either *linear* or *branching*. Mathematically, a strict partial ordering is called *linear* if any two distinct points satisfy the condition: $\forall x, y : x < y \vee x = y \vee x > y$. In linear temporal logics there is only one possible future for each time instant. In case of branching time has a treelike nature as described in the previous section. That is, any particular time instant has one linear past, but many futures. A temporal logic is called *linear time logic* if the structure of time is linear, and *branching time logic* if the underlying semantics of the structure of time is branching. In a linear time logic temporal modalities describe events along a single time line. In contrast, modalities allow quantification over possible futures in branching time logic systems. In both linear and branching time temporal logic systems different logics are obtained by varying the language of the logic. Expressive power changes by constructing the syntax of the logic system from different sets of temporal operators.

As we mentioned above the underlying structure of time in branching time is a tree-like structure. A formal definition of tree can be found in [GHR94]. A *tree* is represented by a tuple $\langle T, < \rangle$, where T is a set of time points and $<$ is a binary ordering relation defined on T. $z \in T$ is called the *root* of a tree $\langle T, < \rangle$ iff for all $x \in T$, $z \leq x$. A *branch* in a tree $\langle T, < \rangle$ is a linearly ordered subset σ of T satisfying that if $x, z \in \sigma$ and $x < y < z$ then $y \in \sigma$. A history in tree $\langle T, < \rangle$ is a maximal linearly ordered subset of T. A tree must satisfy the following conditions:

- $\langle T, < \rangle$ is irreflexive

- $\langle T, < \rangle$ is transitive

- For all $t, u, v \in T$ if $u < t$ and $v < t$ then $u < v$,$u = v$ or $u > v$

- For all $x, y \in T$ there is $z \in T$ such that $z < x$ and $z < y$

A tree is called as a discrete tree if it satisfies the following requirements:

- For all $x \in T$ except the root, there is $y < x$ such that for all $z < x$, $z \leq y$

- For all $x \in T$ there is a set of S of immediate successors of x such that for all $y, z \in S$, neither $y < z$ nor $z < y$ such that for all $z < x$, there is $y \in S$ such that $y \leq z$ (That is, the set S of immediate successors of x is pairwise incomparable)

A discrete tree has branching factor at most σ if and only if each point has at most σ successors.

Temporal logics with underlying branching time structure are fundamental to both computer science and artificial intelligence. Particularly, they have been widely used in AI applications. In planning systems agents formulate different plans and action strategies according to different future world states. Branching time temporal logic is very useful to model the reasoning of agents about the universe of possibilities in which branches represent choices of actions or plans [McD82, RG93].

One of the important applications of branching time temporal logics is formalizing specifications and behaviour of systems. The usage of branching time in specification and verification purposes was proposed in 80s [EC80, Lam80, Abr80]. The unified branching time system UB was defined [BMP81]. Another simple branching time logic, CTL, was introduced in [CE81]. CTL is advantageous in comparison to linear temporal logic, for example, in model checking applications. Model checking with CTL has a linear complexity; whereas model checking with linear temporal logic is of exponential complexity. One criticism on CTL is that it is not sufficiently expressive. These first branching time logics were Peircean in the sense that the truth of formulae was evaluated at points on branching structure [GHR94]. CTL* was introduced in [EH86]. It is an extension over CTL by adding the properties of linear time temporal logic. CTL* is Ockhamist in the sense that truth of formulae is evaluated at points on paths. PCTL*, an extension over CTL*, was introduced in [LS94]. CTL and CTL* include only future time temporal connectives. In contrast, PCTL* contains both past and future time temporal connectives. It should be noted that adding past operators do not increase expressive power; but expressing some properties become easier.

2.2 Choice between Time Instants and Intervals

Temporal ontology, in particular the choice between time instants and time intervals, has been a primary concern in philosophy. Until the last decade most logicians worked on point-based temporal logics. Prior and Pnueli, who developed modal temporal logic in formal philosophy and applied it to system specification and verification respectively, considered time as discrete sequence of points. Formulas in dynamic logic were also interpreted over time instants [Pra76]. The difficulties associated with modeling the refinement of a system specification using a point-based temporal logic are widely recognized as an important problem [FM94].

The interval-based scheme has turned out provide us with a richer representation formalism than the point-based approach. Especially in AI applications the notion of interval is necessary to represent continuous processes and to make temporal statements which are based on intervals.

The concept of time intervals was first studied by Walker [Wal47]. Walker considered a non-empty set of intervals, which is ordered by a partial ordering relation. However, his work does not cover aspects of temporal logic in a general sense. In [Ham71] interval ontology was analyzed philosophically. In [Hum79] an interval tense logic which is based on sub-interval relations was introduced. Dowty emphasized that human language and reasoning have an interval-based semantics rather than point-based one, and he worked on interval-based temporal languages [Dow79]. Similar works in natural languages, such as axiomatic systems for interval-based temporal logics, persistency, homogeneity, were done by [Kam79, Rop80, Bur82, vB83]. In philosophical logic Simons and Galton proposed the requirement of intervals with reference to conceptual structures in natural language [Gal84, Sim87]. Interval-based temporal logics have played an important role in reasoning in artificial intelligence. Some important research has been done within this field by Allen [All83, All84, AH85, AH89, AF94]. This work mainly includes thirteen interval relations, known as Allen's relations, axiomatization and representation of interval structures, and interval-based theory of actions and events. Ladkin worked on completeness theorem and satisfiability algorithms for Allen's logic [Lad87]. Galton pursued a further study on Allen's works [Gal90]. Interval based-logics have been applied to other fields in computer science. In [Par78, Pra79, HPS83] some work on process logic can be found. In process logic intervals represent pieces of information. Another important work was the development of *interval temporal logic*, and its application to design of hardware components [Mos83, HMM83]. ITL had an important impact in temporal logic studies. Various variations have been proposed so far. In particular, Duration Calculus was developed in order to design real-time system specifications formally. Duration Calculus is an extension of interval temporal logic with a calculus to specify and reason about properties of state durations. It uses real numbers to model time, and Boolean-valued functions over time to model states of real-time systems. Some of the articles published in order are [CHR91, HC92, CHS93, CHR93, CX94, Han94, CHX95, HC97, CH98, Cha99, CH04]. Another extension of ITL was developed, which is called *graphical interval logic* that is the foundation of a tool set supporting formal specification and verification of concurrent software systems [Dillon et.

all 1994]. Temporal logic has also been successfully applied in model checking techniques and tools [CG96, PPH98].

In philosophy there are two different perspectives of the structure of an interval type. Intervals are defined in terms of points, which are the only primitive objects, or they are primitive objects in the logic. Most of the interval-based logics construct intervals out of points; there are examples of the latter, though (e.g. Allen's logic [All83]).

2.3 Choice between Propositional and First-order

Propositional temporal logic corresponds to the classical propositional logic. The generic language includes the set of propositional letters, the classical propositional connectives, $\neg$, $\vee$ and $\wedge$, and a set of temporal operators (The propositional constants $\top$ and $\perp$ can also be defined). Propositional temporal logics have the *finite model property* [1] making them useful for the derivation of programs from formal specifications. The derived model resembles a finite state machine; however, the model accepts infinite strings.

Similarly, first-order temporal logic corresponds to predicate logic. Various types of FOTL have been proposed; but the generic language compromises predicate symbols, variables, constants, boolean connectives, quantifiers and temporal operators. FOTL systems can be classified according to several respects. A classification based on the assumption on the structure distinguishes between *uninterpreted* FOTL, where there is a specific domain for each variable and function symbols are partially or fully interpreted, and *interpreted* FOTL where a specific structure is assumed. Another distinction can be made by allowing or disallowing restrictions on the interaction of quantifiers and temporal operators. Some freedom might yield undecidable logics. For example, allowing modal operators within the scope of quantifiers has a severe problem in this sense. In contrast one can disallow such quantification over temporal operators to get a restricted FOTL consisting of propositional temporal logic plus a firstorder language for specifying the atomic propositions [Eme95].

2.4 Choice between Discrete and Dense

The choice between time *discrete* and *dense* time has been also another primary concern in philosophy. If the time is *discrete* then it would consist of a series of instances. Each non-final point is followed by a next point or an immediate successor. Hence we can talk about a property being true in the next instant as well as for all time or at some future time. This can be formulated in first-order logic: $\forall x,y\ (x < y \rightarrow \exists z\ (x < z \wedge z \leq y \wedge \forall w(x < w \wedge w \leq y \rightarrow z \leq w)))$. In most temporal logics used for program reasoning, time is discrete where the present instant corresponds to the program's current state and the

[1] By *finite model property* we mean that if a formula ϕ is satisfiable, then it is satisfiable in a finite model whose size is bounded by some function of the length of the formula ϕ.

next instant corresponds to the program's immediate successor state. Thus the temporal structure corresponding to a sequence of states of a program execution is the nonnegative integers.

A linear ordering is called *dense* if between any two distinct points we can find another different point. This can be mathematically represented as $\forall x,y\ (x < y \rightarrow \exists z\ (x < z < y))$. Rational or the real numbers is quite convenient to represent the flow of dense time, therefore to model the notion of movement. It is noteworthy to mention that there is a distinction between *density* and *continuity*. Suppose that the set of rational numbers is cut into a left and a right half, of numbers smaller and bigger than $\sqrt{2}$, respectively. Such a cut, without a proper point on either edge, is called a gap, and a flow of time is called continuous if it has no gaps. $\mathbb{Q}$ thus forms the standard counterexample, whereas $\mathbb{R}$ and $\mathbb{Z}$ are continuous [Ven98]. Tense logics interpreted over a dense time structure have been investigated by philosophers. Their application to reasoning about concurrent programs was proposed in [BKP86]. Such dense time temporal logics may also have applications in real time programs where strict, quantitative performance requirements are placed on programs.

2.5 Choice between Past and Future

Logicians have provided temporal modal operators to describe the occurrence of events both in past and future. For example, PCTL* is an extension to CTL* and includes both past and future time temporal connectives. Some of temporal logics, e.g. CTL and CTL*, only provide future tense operators. In temporal logic systems for reasoning about concurrency past tense operators do not increase the expressivity since program executions have a definite starting time. For this reason, these logic systems generally do not involve future tense operators. Meanwhile, past tense operators appear to play an important role in compositional specification somewhat analogous to that of history variables [Eme95]. There are some examples showing that use of the past tense operators might be useful simply in order to make the formulation of specifications more natural and convenient (e.g. [LPZ85]).

3 Propositional Temporal Logics

An important success in propositional temporal logic was the introduction of the temporal operators into linear-time temporal logic by Kant [Kam68]. In [GPSS80] a linear temporal logic over discrete time models with *next* and *until* temporal operators was introduced. The models are infinite sequences of states with a first state, but no last state. A sound and complete axiomatic system for propositional temporal logic is provided in [GPSS80]. It was also shown that the logic is decidable and complete. In [LPZ85] the logic of [GPSS80] was extended with the past operators, and a complete proof system for both future and past operators was presented.

In this section we give a general definition of Propositional Temporal Logic, containing future and past operators. A detailed discussion can be found in [Sza95, LP00]. The generic language of propositional temporal logic includes the set of propositional letters Φ, the propositional constants $\top$ and $\bot$, the classical propositional connectives and temporal operators. Formulae of PTL is defined recursively as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi A \psi \mid \phi B \psi$$

where $p \in \Phi$, ϕ and ψ are PTL formulae, A and B are temporal operators.

Let $\mathcal{M} = \langle T, <, \mathcal{V}_{\mathrm{p}} \rangle$ be a model, where T is set of integer time points, $<$ is precedence relation and $\mathcal{V}_p : \Phi \mapsto \{true, false\}$ is a valuation function. The formal semantics of PTL formulas is defined as follows:

$\mathcal{M}, t \models p$ iff $\mathcal{V}_p(t) = \text{true}$ for $p \in \Phi$

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

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

$\mathcal{M}, t \models \phi A \psi$ iff $(\exists t' > t)\ \mathcal{M}, t' \models \phi \wedge \psi$ and $(\forall t'' : t < t'' < t')\ \mathcal{M}, t'' \models \neg\psi$

$\mathcal{M}, t \models \phi B \psi$ iff $(\exists t' < t)\ \mathcal{M}, t' \models \phi \wedge \psi$ and $(\forall t'' : t' < t'' < t)\ \mathcal{M}, t'' \models \neg\psi$

PTL defined here can deal with monadic first-order theory of the flow of time. That is, unary temporal predicates over time and quantifiers over time points can be defined. The definition includes only the operators A and B, which are fully expressive under the integer time structure. This means that, for any monadic first-order formula φ of the flow of time there is a formula φ' of PTL with the same truth table of φ. Additional future and past temporal operators can be defined as follows:

$\overrightarrow{\Diamond}\phi \equiv (true)A\phi :$ *eventually*

$\overleftarrow{\Diamond}\phi \equiv (true)B\phi :$ *once (at some time in the past)*

$\overrightarrow{\Box}\phi \equiv \neg\overrightarrow{\Diamond}\neg\phi :$ *henceforth*

$\overleftarrow{\Box}\phi \equiv \neg\overleftarrow{\Diamond}\phi\neg\phi :$ so *far*

$\overline{\Box}\phi \equiv \overleftarrow{\Box}\phi \wedge \phi \wedge \overrightarrow{\Box}\phi$: *always*

$\overline{\Diamond}\phi \equiv \neg\overline{\Box}\neg\phi :$ *sometimes*

$\phi U \psi \equiv \psi A(\phi \rightarrow \psi) :$ *until*

$\phi S \psi \equiv \psi B(\phi \rightarrow \psi) :$ *since*

$\overrightarrow{\bigcirc}\phi \equiv \phi A(true)$: *immediately after*

$\overleftarrow{\bigcirc}\phi \equiv \phi B(true)$: *immediately before*

It should be noted that the definitions of temporal operators consider *strict past* and *future*, without *now*. Semantics of the logic concern time with a structure of nonnegative integers and a definite starting point.

If a PTL formula is satisfiable then it is satisfiable by a finitely representable model $\mathcal{M}$, whose size can be calculated from the size of the formula.

The theorem can be proved using a standard technique of filtration (see [Gol87]). Since PTL is decidable, there is an algorithm to decide whether a given formula is satisfiable. An automata theoretic technique of obtaining satisfiability procedures for temporal logics can be found in [SVW87].

In the literature several examples of properties of programs expressible by means of temporal logics can be found [Kro87, MP81a, MP81b]. Some important properties are expressed in PTL as follows:

- $p \rightarrow \overrightarrow{\Box} q$ *(safety property)*: All states reached by a program after the state satisfying p do satisfy q.

- $\overrightarrow{\Box}((\neg q) \vee (\neg p))$ *(safety property)*: The program cannot enter critical regions p and q simultaneously.

- $p \rightarrow \overrightarrow{\Diamond} q$: *(liveness property)*: There is a state reached by a program after the state satisfying p does satisfy q.

- $\overrightarrow{\Box}\overrightarrow{\Diamond} p \rightarrow \overrightarrow{\Diamond} q$: *(liveness property)*: If a request p is repeated, a response q is received.

- $\overrightarrow{\Box} p \rightarrow \overrightarrow{\Diamond} q$ *(liveness property)*: If a request p is hold permanently, a response q is received.

In the literature there are two different approaches to time models: one is fully symmetric time models, infinite in both directions, and the other is the models with definite starting point and the corresponding restriction to future temporal operators. The restriction to models with definite starting point is done by describing the computations of programs which all have an initial state. If the logic with past operators over linear models are considered with a first state, there are two different ways of defining the notion of satisfiability of a formula φ by a model $\sigma : s_0, s_1, \ldots$. The first interpretation is the *floating interpretation* by which $\sigma \models \varphi$ if φ holds at every state of σ. The second is the *anchored interpretation*, by which $\sigma \models \varphi$ if φ holds at the first state of σ. In [LPZ85] the former approach was used. Later it has been found that the latter approach is better to properly define the temporal hierarchy.

In [LP00] a collection of the recent developments was presented, by presenting a propositional temporal logic with the full set of operators under the *anchored interpretation*, proposing an axiomatic system for this logic and establishing decidability of the logic and completeness of the axiomatic system by an improved tableau method. The logic proposed is a complete PTL with the past temporal operators. The main temporal operators are given below. Additional temporal operators, including *weak until* and *weak since*, can also be defined as showed above: $\bigcirc$ - *next, U- until,* $\overline{\bigcirc}$- *weak previous* and S- since.

In [SC85] it was found that the satisfiability problem for PTL is *PSPACE-complete*. In [LP00] an exponential time tableau algorithm, which has an overall time bound of $2^{O(|\phi|)}$, is presented for the validity problem for PTL. The idea is that the negation of the formula is checked for satisfiability, using a semantic tableaux. It is then decomposed into its subformulas according to the different possibilities that come from disjunctions. This decomposition is progressing in a tree-like tableaux, where each branch is terminated either by a leaf that contains some subformula and its negation, which means that this branch does not present a model, or by a leaf that contains a set of non-contradictory propositions or negations of propositions, that represent some model.

A complete deductive system for proving the validity of PTL formulas is also provided. The presented formal proof for completeness uses standard method for showing the completeness of a deductive system: If φ is a valid PTL formula, then to show that $\vdash \varphi$, the process of checking satisfiability of the formula $\neg\varphi$ over its semantic tableaux is investigated.

Among the proof systems existing in literature are Hilbert-style proof system [Lad87] and Gentzen-style proof system [Sza95]. Both proof systems are sound and complete.

4 First-Order Temporal Logics

First-order temporal logic is an extension of propositional temporal logic. Besides all features of propositional temporal logic it also allows arbitrary data structures and quantifiers over individuals. First-order temporal logic systems have found numerous applications in computer science and artificial intelligence: A typical application is their usage in specification and verification of reactive systems. They provide more expressive and powerful tools for formalising the behaviour of executable temporal logics [Fis96a, Fis96b]. They could extend model checking techniques to non finite-state systems, and to systems containing multiple concurrent processes [Hol91, CGP00]. They also allow the extension of techniques for reasoning about knowledge to more dynamic and powerful classes [FHMV96, WZ00]. First-order logic systems are also quite useful in information systems in the sense that query languages for temporal databases are often based on variants of FOTL [CT98]. In addition, they provide a means for verifying properties of transaction protocols in e-commerce [AVFY98, Spi00].

Despite its usefulness in various areas, first-order temporal logic is a very expressive language with a very high computational complexity. Although some axiomatizations of first-order temporal logics have been studied [Rey96], many varieties of FOTL are not even recursively enumerable [Aba89, ANS79, GHR94, Mer92], and so do not admit finite proof methods at all. For that reason, most of the works in this field have mainly dealt with developing PTL-based tools. There are very few examples that recursively enumerable or decidable fragments of first-order temporal logics have been found. However, these variations were just small extensions to the propositional case. Some examples of these extensions are weaker versions of validity [Aba89], minimal interaction between quantifiers and temporal operators [Cho95] and very restricted first-order features [Mer92, Pli97].

One important development was done by Hodkinson, Wolter, and Zakharyaschev, who introduced a new natural *monodic* fragment of first-order temporal logic, and showed that it is quite expressive and have much better computational behaviour [HWZ00]. In monodic formulas, the scope of temporal operators is restricted only to subformulas with at most one free variable. The whole monodic fragment can be represented as a finite axiomatic system [WZ02], and so can be supported by tableau or resolution-type reasoning mechanism. Moreover, by restricting the first-order part to certain decidable fragments, decidable monodic fragments of first-order temporal logic can be obtained over various flows of time [ANvB98, Gra99].

In this section we will give a general definition of first-order temporal logic defined in [HWZ00]. Thereafter, we will show some decidable and undecidable fragments.

The logic first-order temporal is represented by QTL, which is constructed from the following alphabet, which does not comprise equality and function symbols:

predicate symbols: $P_0, P_1 ...$;

variables: $x_0, x_1, ...$;

constants: $c_0, c_1, ...$;

boolean connectives: $\wedge$, $\neg$;

universal quantifier: $\forall$;

temporal operators: S(*since*) and U(*until*);

Additional temporal operators can be defined as a similar to PTL.

Let $\mathcal{M} = \langle \mathbb{T}, \mathcal{D}, \mathcal{I} \rangle$ be a first-order temporal model where $\mathbb{T} = \langle T, < \rangle$ is a strict linear order representing time, $\mathcal{D}$ is a non-empty domain set of $\mathcal{M}$, and $\mathcal{I}$ is a function assigning a first-order structure of the form

$$\mathcal{I}(t) = \left\langle \mathcal{D}, P_0^{\mathcal{I}(t)}, ..., c_0^{\mathcal{I}(t)}, ... \right\rangle$$

to every $t \in T$. For every i, $P_i^{\mathcal{I}(t)}$ is a predicate on $\mathcal{D}$ of the same arity as P_{i}. The formal semantics of QTL is defined as follows:

$\mathcal{M}, \alpha, t \models P_i(x_1, ..., x_n)$ iff $P_i^{\mathcal{I}(t)}(\alpha(x_1), ..., \alpha(x_n))$ is true in $\mathcal{I}(t)$, where x_i are variables or constants

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

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

$\mathcal{M}, \alpha, t \models \phi x \forall$ iff $\mathcal{M}, \beta, t \models \phi$ for any valuation β which differs from α at most in the value of variable x

$\mathcal{M}, \alpha, t \models \phi S \psi$ iff $(\exists t' < t)\ \mathcal{M}, \alpha, t' \models \psi$ and $(\forall t'' : t' < t'' < t)\ \mathcal{M}, \alpha, t'' \models \phi$

$\mathcal{M}, \alpha, t \models \phi U \psi$ iff $(\exists t' > t)\ \mathcal{M}, \alpha, t' \models \psi$ and $(\forall t'' : t < t'' < t')\ \mathcal{M}, \alpha, t'' \models \phi$

where α and β are valuation functions which assign values from $\mathcal{D}$ to variables.

Let F be the underlying time structures assumed for QTL defined here constitutes strict linear orders. Then, $QTL(F)$ denotes the first-order temporal logic of F, and $QTL_{fin}(F)$ denotes the logic of F with *finite domains.*

4.1 Undecidable Fragments of QTL

In the literature, it has been known that both the monadic and two-variable fragments of classical first-order logic are decidable [BGG97]. However, the computational complexities of their temporal counterparts are different. Let QTL^2 denote the *two - variable fragment* of QTL, and QTL^{mo} denote the *monadic fragment* (not monodic) of QTL, which respectively means that every formula in QTL^2 contains at most 2 distinct individual variables, and the set of formulas that contain only unary predicates and propositional variables. The theorems below show the complexities of these two fragments of QTL.

Let $\mathbb{T}$ be either $\{\langle\mathbb{N}, <\rangle\}$ or $\{\langle\mathbb{Z}, <\rangle\}$. Then $QTL^2 \cap QTL^{mo} \cap QTL(\mathbb{T})$ is not recursively enumerable.

The theorem can be proved by reducing the recurrent tiling problem for $\mathbb{N} \times \mathbb{N}$, which is $\sum_1^1$- complete, to the satisfiability problem for the monadic QTL^2 - formulas (see [HWZ00]).

Let F be either $\{\langle\mathbb{N}, <\rangle\}$ or $\{\langle\mathbb{Z}, <\rangle\}$. Then $QTL^2 \cap QTL^{mo} \cap QTL_{fin}(F)$ is not recursively enumerable.

The theorem can proved by reducing the problem of whether a Turing machine comes to a stop having started from the empty tape, which is known to be undecidable, to the satisfaction problem for monodic QTL^2 - formulas in models with finite domain (see [HWZ00]).

4.2 Decidable Fragments of QTL

In the theorems given above there is a quantification in three 'dimensions', one temporal and two domain, since the linear time operator U can be applied to formulas with two free variables. This causes a problem that these fragments of QTL are undecidable. It is known that the three-variable fragment of classical first-order logic is undecidable [BGG97].

In order to avoid this problem corresponding fragment of QTL, which is QTL_1, contains all QTL-formulas φ such that any subformula of φ of the form $\psi_1 U \psi_2$ and $\psi_1 S \psi_2$has at most one free variable. These formulas are *monodic* (not monadic) formulas, allowing quantification into temporal contexts only with one free variable. The monodic fragments of $QTL(\langle \mathbb{N}, < \rangle)$ and $QTL(\langle \mathbb{Z}, < \rangle)$ are recursively enumerable.

Let F be any of the following classes of flows of time: $\{\langle \mathbb{N}, < \rangle\}$, $\{\langle \mathbb{Z}, < \rangle\}$, $\{\langle \mathbb{Q}, < \rangle\}$ the class of all finite strict linear orders, any first-order-definable class of strict linear orders, and F^+ be F and $\{\langle \mathbb{R}, < \rangle\}$. Then, the following fragments are decidable *:* $QTL(F) \cap QTL^1$, $QTL(F) \cap QTL_1^2$, $QTL(F) \cap QTL_1^{mo}$, $QTL_{fin}(F^+) \cap QTL^1$, $QTL_{fin}(F^+) \cap QTL_1^2$, $QTL_{fin}(F^+) \cap QTL_1^{mo}$.

The proof is based on *quasimodels.* A quasimodel of φ over a flow of time $\mathbb{T} = \langle T, < \rangle$ comprises as assignment of a realizable set of types S_t to each $t \in T$, the sequence $\langle S_t : t \in T \rangle$ having certain specified properties. It can be shown that φ is satisfiable in a model $\mathbb{T}$ iff there exists a quasimodel for φ over $\mathbb{T}$ (see [HWZ00]).

In [GKWZ02] it was shown that $QTL(\langle \mathbb{N}, < \rangle) \cap QTL^1$ is *EXPSPACE-hard.* It also follows that the satisfiability problem for QTL_1^{mo}-formulas in models based on $\langle \mathbb{N}, < \rangle$ is *EXPSPACE-complete.*

It has been assumed in this section that QTL and its fragments do not include *equality* and *function symbols.* It can be shown that undecidability is a major problem with the logic extended with function symbols [WZ02]. For example, the set of one-variable formulas with one function symbol that are valid in models based on $\langle \mathbb{N}, < \rangle$ is not recursively enumerable. Moreover, the set of monodic QTL formulas with equality that are valid in all temporal models based on $\langle \mathbb{N}, < \rangle$ is not recursively enumerable. In [WZ02] a finite *Hilbert-style axiomatization* of monodic fragment of first-order temporal logic was constructed. It was also proved that the monodic fragment with equality is not recursively axiomatizable.

The decidability results can be extended to temporalized description logics. The resulting temporalized description logics are suitable for temporal conceptual modelling. These recent research results have showed that relatively expressive subsets of first-order temporal logic could be found. In [WZ99] certain similarities between monodic first-order temporal logic and effective multi-dimensional knowledge representation formalisms are described and it has been suggested that the monodic first-order temporal logic systems can be considerably extended. In [HWZ00, WZ02] there is a scope for enriching the expressive power of the monodic fragment. For example, applications of temporal operators, such as next-time, can be allowed to formulas with two or more free variables. Formulas of this form are particularly useful in temporal databases, and cover the decidable fragments of first-order temporal logic developed by Pliuskevicius [Pli97]. Some recent works present tableau-based satisfiability checking algorithms for description logics with temporal and epistemic operators [LSWZ02]. Similarities between such logics and monodic temporal logics suggest that tableau-based reasoning systems can also be constructed for decidable monodic fragments. This can be carried out by combining existing tableau systems for PTL and the classical first-order components.

5 Branching Time and Partial Order Temporal Logics

5.1 Branching Time Temporal Logics

A temporal logic system is called *branching time logic* if the underlying semantics of the structure of time is branching. Underlying structure of time in branching time is a tree-like structure. That is, every time instant may have several immediate successors which correspond to different futures. A formal definition of branching time structure has been given in Section 2.

Temporal logics with underlying branching time is fundamental to both computer science and artificial intelligence. Particularly, it has been widely used in AI applications. In planning systems agents formulate different plans and action strategies according to different future world states. Branching time temporal logics are very useful to model the reasoning of agents about the universe of possibilities in which branches represent choices of actions or plans [McD82, RG93]. Another important applications of these logics is formulizing specifications and behaviour of systems.

The first ideas about branching time logics appeared in [Abr80]. Later, The unified branching time system UB was defined [BMP81]. A simple branching time logic, CTL, was introduced in [CE81]. Thereafter, CTL* was introduced in [EH86]. CTL* is an extension over CTL by adding the properties of linear time temporal logic. PCTL*, an extension over CTL*, was introduced in [LS94]. UB, CTL and CTL* include only future time temporal connectives. In contrast, PCTL* contains both past and future time temporal connectives.

In this section the syntax, semantics, expressiveness and characterizations of some of the branching time logics are briefly represented.

5.1.1 *Computational Tree Logic (CTL)*

CTL is an extension to the logic UB by adding a new path modality U. CTL formulas are recursively defined as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \forall\,(\phi U \psi) \mid \exists\,(\phi U \psi) \mid \exists\,(\bigcirc\phi)$$

where p is a set of atomic propositions Φ, ϕ and ψ are CTL formulas, $\bigcirc$ and U are linear time operators, and $\exists$ and $\forall$ are path quantifiers. $\Diamond$, $\Box$ and other modalities can be derived as usual.

Although branching time logics can be defined in a tree-like structure, in many applications it is quite useful to define the logic in a different way than the tree structure. In this approach a set of states with a transition relation are considered as the basic object. The language of CTL contains states formulas only.

Let $\mathcal{M} = \langle S, R, \mathcal{V} \rangle$ be a model , where S is nonempty set of state, R is a binary relation $R \subseteq S \times S$ and $\mathcal{V} : S \rightarrow 2^{\Phi}$ is a valuation function which assigns to each state a set of atomic propositions. The formal semantics of CTL is then defined as follows:

$\mathcal{M}, s_0 \models p$ iff $p \in \mathcal{V}(s_0)$ for $p \in \Phi$

$\mathcal{M}, s_0 \models \neg\phi$ iff not $\mathcal{M}, s_0 \models \phi$

$\mathcal{M}, s_0 \models \phi \wedge \psi$ iff $\mathcal{M}, s_0 \models \phi$ *and* $\mathcal{M}, s_0 \models \psi$

$\mathcal{M}, s_0 \models \exists\,(\phi U \psi)$ iff there is a forward fullpath $\sigma = s_0, s_1, ...$ such that $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i) \mathcal{M}, s_j \models \phi$

$\mathcal{M}, s_0 \models \forall\,(\phi U \psi)$ iff for all forward fullpaths $\sigma = s_0, s_1, ...$ $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i)\ \mathcal{M}, s_j \models \phi$

$\mathcal{M}, s_0 \models \bigcirc\phi$ iff $\mathcal{M}, s_1 \models \phi$

$\mathcal{M}, s_0 \models \phi$ denotes that the state formula ϕ is true at the state s in $\mathcal{M}$. A *forward fullpath* of $\langle S, R \rangle$ is an infinite sequence $s_0, s_1, s_2, ...$ with each $\langle s_i, s_{i+1} \rangle \in R$. σ_i denotes the suffix $s_i, s_{i+1} ...$ of σ. A tree can be obtained by looking at all the finite sequences $s_0, s_1, s_2, ..., s_m$ which begin with a particular state s_0 and go on satisfying $\langle s_i, s_{i+1} \rangle \in R$. The set of such finite sequences can be ordered by prefixing $s_0, s_1, s_2, ..., s_m < s_0, s_1, s_2, ..., s_m, s_{m+1}, ..., s_n$.

In [Pen95] a sound and complete axiomatic system is provided for CTL. It has to be shown that any consistent formula ϕ is satisfiable. The proof rests on constructing of a pseudo-Hintikka structure for a satisfiable CTL formula ϕ.

It can be shown that CTL has the finite model property. That is, if a formula ϕ is satisfiable, then it is satisfiable in a finite model whose size is bounded by some function of the length of the formula ϕ. Hence, a non-deterministic algorithm can determine the satisfiability of a CTL formula in polynomial time. Therefore, CTL is decidable [EH82]. There is also a tableau-based deterministic exponential time complete procedure for CTL satisfiability [EC82].

It is noteworthy to mention that the logic UB has the finite model property, as well. It has a sound and complete axiomatization system, and there is a deterministic exponential time lower bound for UB satisfiability.

5.1.2 *Full Computational Tree Logic (CTL*)*

There are two kinds of CTL* formulas: *state* formulas and *path* formulas. State formulas are interpreted over states and path formulas, containing all state formulas, are interpreted over paths. CTL* formulas are recursively defined as follows:

$$\alpha ::= p \mid \neg\alpha \mid \alpha \wedge \beta \mid \exists\phi$$

$$\phi ::= \alpha \mid \neg\phi \mid \phi \wedge \psi \mid \phi U \psi \mid \bigcirc\phi$$

where $p \in$ a set of atomic propositions Φ, ϕ and ψ are path formulas, α and β are state formulas, $\bigcirc$ and U are path modalities, and $\exists$ is path quantifier. $\Diamond$, $\Box$ and other modalities can be derived as usual.

Let $\mathcal{M} = \langle S, R, \mathcal{V} \rangle$ be a model , where S is nonempty set of state, R is a binary relation $R \subseteq S \times S$ and $\mathcal{V} : S \to 2^{\Phi}$ is a valuation function. The formal semantics of CTL* is then defined as follows:

$\mathcal{M}, s \models p$ iff $p \in \mathcal{V}(s)$ for $p \in \Phi$

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

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

$\mathcal{M}, s \models \exists\phi$ iff $\mathcal{M}, \sigma \models \phi$ for some fullpath σ starting at s

$\mathcal{M}, \sigma \models \phi$iff $\mathcal{M}, s_0 \models \phi$for any state formula ϕ

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

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

$\mathcal{M}, \sigma \models \bigcirc\phi$ iff $\mathcal{M}, \sigma_1 \models \phi$

$\mathcal{M}, \sigma \models \phi U \psi$ iff $(\exists i \geq 0)$ $\mathcal{M}, \sigma_i \models \psi$ and $(\forall j : 0 \leq j < i)$ $\mathcal{M}, \sigma_j \models \phi$

$\mathcal{M}, s \models \phi$ denotes that the state formula ϕ is true at the state s in $\mathcal{M}$. Similarly, $\mathcal{M}, \sigma \models \phi$ denotes that the path formula ϕ is true in the fullpath σ in $\mathcal{M}$.

The axiomatizability of CTL* was an open question for a long time. A sound and complete axiomatization for CTL* has recently been defined by Reynolds in [Rey01].

There is an algorithm to decide the satisfiability of CTL* formula, which has a double exponential complexity in the length of the formula.

The idea behind the proof is to find a tree automaton which accepts the models of the given CTL* formula exactly, to find a deterministic equivalent and to check whether that is empty. That is, the satisfiability problem is reduced to testing the nonemptiness of tree automata [EJ88].

5.1.3 *Full Computational Tree Logic with Past (PCTL*)*

As it mentioned before there are two ways in formalizing branching time temporal logics. In the semantics definitions CTL and CTL* we used states and paths as basic object types. In this section we will use discrete ω-height branching which is equivalent to state representation. That is, PCTL* formulas are evaluated at nodes on branches of labelled discrete rooted trees of height ω. Whereas the language of CTL* contains state and path formulas, some formulas of PCTL* do not depend on the path on which they are evaluated. PCTL* formulas are recursively defined as follows :

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi U \psi \mid \phi S \psi \mid \Diamond \phi$$

where p is a set of atomic propositions Φ, $\phi$are ψ PCTL* formulas, and U, S and $\Diamond$ are path modalities.

Let $\mathcal{M} = \langle T, <, h \rangle$ be a model , where T is a set of time points ordered by a binary relation $<$. The formal semantics of PCTL* is then defined as follows *[GRF00]:*

$\mathcal{M}, x, \sigma \models p$ iff $t \in h(p)$

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

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

$\mathcal{M}, x, \sigma \models \phi U \psi$ iff there is a node $y > x$ in the branch σ such that $\mathcal{M}, y, \sigma \models \phi$ and for all nodes z , $x < z < y$, $\mathcal{M}, z, \sigma \models \psi$

$\mathcal{M}, x, \sigma \models \phi S \psi$ iff there is a node $y < x$ in the branch σ such that $\mathcal{M}, y, \sigma \models \phi$ and for all nodes z , $y < z < x$, $\mathcal{M}, z, \sigma \models \psi$

$\mathcal{M}, x, \sigma \models \Diamond\phi$ iff there is a branch τ containing x such that $\mathcal{M}, x, \tau \models \phi$

$\mathcal{M}, x, \sigma \models \phi$ denotes that ϕ holds at the node x on the branch σ of model $\mathcal{M}$.

Addition of past operator to the language does not increase expressive power. Decidability of PCTL* follows directly from the expressibility observation along with the decidability of CTL*. Until this year the axiomatizability of PCTL* has been a long-lasting open question. Reynolds gives a sound and complete axiomatizion system for PCTL* in [Rey05].

5.1.4 *Expressiveness of Branching Temporal Logics*

One of the main use of branching time logics in computer science is that the model-checking procedure is very efficient. The task is to represent a given system as a Kripke structure and check that it is a model of a given specification. CTL is quite adequate to express a certain set of useful properties. In contrast to the exponential complexities of model checking with a linear temporal logic, model checking with CTL is of linear complexity. Model checking with CTL* is much more complex than CTL, which is *PSPACE-complete*, as it needs a recursion involving checking of all paths from a particular state [GRF00].

The branching logic systems can also be used to specify properties of concurrent programs. A frame of the logic represents an execution tree generated by a program. The system properties which can be expressed by means of UB are as follows:

$\forall\Box p$: *safety property*: p is true at all states of each path.

$\forall\Diamond p$: *liveness property*: p is true at some state of each path.

$\exists\Diamond p$: *possibility property*: p is true at some state of some path.

Fairness constraints are not expressible in UB. All properties expressible in UB are also defined in CTL. The new properties, such as relative order of events, expressed by CTL contains the modality U. As in UB, fairness constraints cannot be expressible in CTL. CTL* can specify more properties over UB and CTL [Pen95]:

$\Box\Diamond p$: *impartiality property*

$\Box\Diamond p \rightarrow \Box\Diamond q$: *fairness property*

$\Diamond\Box p \rightarrow \Box\Diamond q$: *justice property*

$\exists\,((pUq) \vee \Box p)$: *weak until property*

These languages mentioned in this section can be made more expressive, while still keeping all their formulas as state formulas, by allowing classical connectives in between the temporal connectives and the path connective. If we add past operators does not increase expressiveness; it just allows more convenient notation to express useful properties. Due to complexity and expressiveness considerations some other logics have been defined, such as CTL^+[EH82], $ECTL$ [EC82], $ECTL^+$[EC82].

5.2 Partial Order Temporal Logics

Partial order structures are similar to branching structures except that every time instant may also have several immediate predecessors corresponding to different pasts. The first logic based on partial orders was POTL, which is introduced in [PW84], and later its extended version, POTL[U,S], defined in [KP86]. POTL and POTL[U,S] can be viewed as extensions of UB and CTL by past modalities. However, their semantic structures can be linked with partial orderings representing runs of concurrent systems.

5.2.1 *POTL*

POTL is intended to describe partially ordered computations directly. Hence, it is possible to specify that states with several successors and several predecessors. The language of POTL is an extension of the language UB by allowing quantification over backward paths. POTL formulas are recursively defined as follows [Pen95]:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \exists\overrightarrow{\bigcirc}\phi \mid \exists\overrightarrow{\Diamond}\phi \mid \exists\overrightarrow{\Box}\phi \mid \exists\overleftarrow{\bigcirc}\phi \mid \exists\overleftarrow{\Diamond}\phi \mid \exists\overleftarrow{\Box}\phi$$

where $p \in$ a set of atomic propositions Φ, ϕ and ψ are POTL formulas, and $\exists\overrightarrow{\bigcirc}, \exists\overrightarrow{\Diamond}, \exists\overrightarrow{\Box}, \exists\overleftarrow{\bigcirc}, \exists\overleftarrow{\Diamond}, \exists\overleftarrow{\Box}$ are modalities. The duals of the modalities can be defined as usual.

Let $\mathcal{M} = \langle S, R, \mathcal{V}\rangle$ be a model , where S is nonempty set of state, R is a binary relation $R \subseteq S \times S$ and $\mathcal{V} : S \rightarrow 2^{\Phi}$ is a valuation function. It is assumed that each state has at least one R-successor and on R-predecessor. The formal semantics of POTL is then defined as follows:

$\mathcal{M}, s_0 \models p$ iff $p \in \mathcal{V}(s_0)$ for $p \in \Phi$

$\mathcal{M}, s_0 \models \neg\phi$ iff not $\mathcal{M}, s_0 \models \phi$

$\mathcal{M}, s_0 \models \phi \wedge \psi$ iff $\mathcal{M}, s_0 \models \phi$ *and* $\mathcal{M}, s_0 \models \psi$

$\mathcal{M}, s_0 \models \exists\overrightarrow{\bigcirc}\phi$iff there is a forward fullpath $\sigma = s_0, s_1, ...$ s.t. $\mathcal{M}, s_1 \models \phi$

$\mathcal{M}, s_0 \models \exists\overrightarrow{\Diamond}\phi$ iff there is a forward fullpath $\sigma = s_0, s_1, ...$ s.t. $\exists i \geq 0\ \mathcal{M}, s_i \models \phi$

$\mathcal{M}, s_0 \models \exists\overrightarrow{\Box}\phi$iff there is a forward fullpath $\sigma = s_0, s_1, ...$ s.t. $\forall i \geq 0\ \mathcal{M}, s_i \models \phi$

$\mathcal{M}, s_0 \models \exists\overleftarrow{\bigcirc}\phi$ iff there is a backward fullpath $\sigma = ..., s_1, s_0$ s.t. $\mathcal{M}, s_1 \models \phi$

$\mathcal{M}, s_0 \models \exists\overleftarrow{\Diamond}\phi$ iff there is a backward fullpath $\sigma = ..., s_1, s_0$ s.t. $\exists i \geq 0\ \mathcal{M}, s_i \models \phi$

$\mathcal{M}, s_0 \models \exists\overleftarrow{\Box}\phi$ iff there is a backward fullpath $\sigma = ..., s_1, s_0$ s.t. $\forall i \geq 0\ \mathcal{M}, s_i \models \phi$

POTL does not have the finite model property due to the addition of backward operators. The proof is based on showing that the formula $\phi \wedge \forall\overrightarrow{\bigcirc}\forall\overrightarrow{\Box}\neg\phi \wedge \forall\overrightarrow{\Box}\forall\overleftarrow{\Diamond}\phi$ is satisfiable in infinite models only. Checking whether a POTL formula is satisfiable requires an exponential time algorithm. A sound and complete proof system for POTL can be found in [Pen95].

POTL[U,S] is the extension of POTL obtained by introducing *until* and *since*. There is a difference in the definition of POTL[U,S] model that the initial or terminal states of some process may have no R-successors or predecessors, respectively.

The formal semantics of formulas containing until and since is defined as follows:

$\mathcal{M}, s_0 \models \exists(\phi U \psi)$ iff there is a forward fullpath $\sigma = s_0, s_1, ...$ such that $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i)$, $\mathcal{M}, s_j \models \phi$

$\mathcal{M}, s_0 \models \forall(\phi U \psi)$ iff for all forward fullpaths $\sigma = s_0, s_1, ...$ such that $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i)$, $\mathcal{M}, s_j \models \phi$

$\mathcal{M}, s_0 \models \exists(\phi S \psi)$ iff there is a backward fullpath $\sigma = ..., s_1, s_0$ such that $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i)$, $\mathcal{M}, s_j \models \phi$

$\mathcal{M}, s_0 \models \forall(\phi S \psi)$ iff for all backward fullpaths $\sigma = ..., s_1, s_0$ such that $(\exists i \geq 0)\ \mathcal{M}, s_i \models \psi$ and $(\forall j : 0 \leq j < i)$, $\mathcal{M}, s_j \models \phi$

In the semantics of POTL[U,S] the definition of forward and backward fullpaths changes a bit. A forward fullpath is a *maximal* sequence of states $s_0, s_1, s_2, ...$ with each $\langle s_i, s_{i+1} \rangle \in R$. A forward fullpath is finite iff its last state does not have any R-successor. Backward fullpath are defined similarly.

POTL[U,S] extends POTL, and since POTL does not have the finite model property, POTL[U,S] does not have it either. An in POTL the proof is based on showing that the formula $\phi \wedge \forall\overrightarrow{\bigcirc}\forall\overrightarrow{\Box}\neg\phi \wedge \forall\overrightarrow{\Box}\forall\overleftarrow{\Diamond}\phi$ is satisfiable in infinite models only. There is a deterministic algorithm for deciding whether a POTL[U,S] formula is satisfiable, of exponential complexity in the length of the tested formula. A sound and complete axiomatization system for POTL[U,S] is given in [Pen95].

5.2.2 *Expressiveness of Partial Order Temporal Logics*

POTL extends the expressiveness of UB by referring to the past. However there is a difference between UB and POTL frameworks. The structure of the former represents an entire concurrent system. In the latter, a structure represents one possible run of a system composed of sequential processes. In this framework, POTL is used to specify properties involving all runs. For example, $q \rightarrow \forall\overleftarrow{\Diamond} p$ expresses that for every run, and for all backward fullpaths ending at states where q holds, there is a state at which p holds. All properties expressible in POTL can be expressible in POTL[U,S]. In addition, POTL[U,S] allows specifying the properties concerning the relative order of events in the future and past. Model checking with POTL[U,S] is more complicated than CTL. Indeed, the complexity is exponential in the size of the model and doubly exponential in the length of the tested formula [KP86]. The reason for this high complexity is that POTL[U,S] formulas contain backward modalities, and they are interpreted over models corresponding to runs of concurrent systems.

6 Interval Temporal Logics

The interval-based scheme provides us with a richer representation formalism than the point-based approach. In philosophy there are two different perspectives of the structure of an interval type. Intervals are defined in terms of points, which are the only primitive objects, or they are primitive objects in the logic. Most of the interval-based logics construct intervals out of points.

In the former view, an underlying flow of time is modelled as a strict partial ordering of time points, while intervals are defined as sets of time points satisfying some particular constraints.

The latter approach is followed in [vB91], where a "resonable choice of basic principles embodying the minimum conditions for a structure to qualify as a periodic structure" was analysed. The author gives two examples of interval structures: the closed intervals over the intergers and the open intervals over the reals. He then defines general principles by abstracting from those concrete structures. He considers the subinterval relation $\subseteq$ and precedence $\prec$ between intervals, and studies the first-order theory of structures of the form $(I, \subseteq, \prec)$, where I is an interval. The same approach is followed in [MSV02]. The authors define an interval theory of a particular class of interval structures (Split structures) and interval logics (Split Logics).

6.1 More Into Intervals

Given that $\mathbb{T} = \langle T, < \rangle$ is a strict partial ordering and T is a set of time points, an **interval** in $\mathbb{T}$ is a pair $[t_1, t_2]$ such that $t_1, t_2 \in T$. $[t_1, t_2]$ is a **strict interval** if $t_1 < t_2$. $[t_1, t_2]$ is a **non-strict interval** if $t_1 \leq t_2$. Intervals of the form $[t_1, t_1]$are called **point intervals**.

We denote the set of strict intervals on $\mathbb{T}$ as $\mathbb{I}(\mathbb{T})^-$, and the set of all (strict and point) intervals on $\mathbb{T}$ as $\mathbb{I}(\mathbb{T})^+$. $\mathbb{I}(\mathbb{T})$ denotes either of these sets.

Given a strict partial ordering $\mathbb{T} = \langle T, < \rangle$ and a set of intervals $\mathbb{I}(\mathbb{T})$, we call a pair $\langle \mathbb{T}, \mathbb{I}(\mathbb{T}) \rangle$ an **interval structure**. The interval structures $\langle \mathbb{T}, \mathbb{I}(\mathbb{T})^- \rangle$, $\langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+ \rangle$ denote **strict** and **non-strict interval structure**, respectively.

In the literature it is well known that there are *Allen's* thirteen different binary relations between intervals on a linear ordering, which are *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, *equals*, *finished by*, *during by*, *started by*, *overlapped by*, *met by*, *after* [All83].

Another natural binary relation between intervals, definable in terms of the Allen's relations, is the *sub-interval* relation. Given a strict partial ordering $\mathbb{T}$ and two intervals $[t_1, t_2]$ and $[t'_1, t'_2]$, we have that $[t_1, t_2]$ is a *sub-interval* ($\subseteq$) of $[t'_1, t'_2]$ if $t_1 \leq t'_1$ and $t_2 \leq t'_2$; $[t_1, t_2]$ is a *strict sub-interval* ($\subset$) of $[t'_1, t'_2]$ if $t_1 < t'_1$ and $t_2 < t'_2$; $[t_1, t_2]$ is a *proper sub-interval* ($\overline{\subset}$) of $[t'_1, t'_2]$ if $[t_1, t_2]\subseteq[t'_1, t'_2]$ and $[t_1, t_2]\neq[t'_1, t'_2]$.

In the area of propositional interval logics another important relation is *ternary* relation A between intervals [Ven91]. For given intervals i, j, k $Aijk$ holds if and only *if i meets j, i begins k,* and *j ends k.*

6.2 Propositional Interval Temporal Logics

In this section we will present well-known propositional interval logics, which involve unary or binary modal operators, and whose semantic structures are over partial orderings with linear interval property, i.e. orderings in which every interval is linear (see [GMS04]).

The generic language of propositional interval temporal logics includes the set of propositional variables Φ, the propositional constants $\bot$ and $\top$, the Boolean connectives, and a set of modal operators specific to each logical system.

There are two different natural semantics for interval temporal logics. A *strict model* for a formula is a tuple $\mathcal{M}^- = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^-, \mathcal{V} \rangle$ where $\langle \mathbb{T}, \mathbb{I}(\mathbb{T})^- \rangle$ is a strict interval structure with linear interval property, and $\mathcal{V} : \Phi \mapsto 2^{\mathbb{I}(\mathbb{T})^-}$ is a *valuation function* that which associates each propositional variable p with the set of intervals where p is true. A *non-strict model* is a tuple $\mathcal{M}^+ = \langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+, \mathcal{V} \rangle$ where $\langle \mathbb{T}, \mathbb{I}(\mathbb{T})^+ \rangle$ is a non-strict interval structure with linear interval property, and $\mathcal{V} : \Phi \mapsto 2^{\mathbb{I}(\mathbb{T})^+}$. $\mathcal{M}$ denotes either of these models.

6.2.1 *The Logic HS*

The logic HS [HS91] is one of the most expressive propositional interval temporal logics that have been defined so far. The formulas of the logic HS are recursively defined as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle A \rangle \phi \mid \langle B \rangle \phi \mid \langle E \rangle \phi \mid \langle \bar{A} \rangle \phi \mid \langle \bar{B} \rangle \phi \mid \langle \bar{E} \rangle \phi$$

Allen's all relations between two distinct intervals can be expressed by means of these modal operators. The informal semantics are as follows:

$\langle A \rangle \phi$: ϕ holds at some interval that begins just after the current interval ends.

$\langle B \rangle \phi$: ϕ holds at some interval that begins with the current interval, and ends before it ends.

$\langle E \rangle \phi$: ϕ holds at some interval that begins after the current interval starts, and ends when it ends.

$\langle \bar{A} \rangle \phi$: ϕ holds at some interval that ends just before the current interval starts.

$\langle \bar{B} \rangle \phi$: ϕ holds at some interval that begins with the current interval, and ends after it ends.

$\langle\bar{E}\rangle\,\phi$: ϕ holds at some interval that begins before the current interval starts, and ends when it ends.

The duals of the modalities can be also be defined: $[X\phi] \equiv \neg\langle X\rangle\neg\phi$, where X represents the modal operators. Other modalities, such as $\langle D\rangle$, $\langle L\rangle$, $\langle O\rangle$ etc., and their duals can be formalized in terms of the modalities given above. If we define modal constant π as $[B]\perp$, the begin and end of the current interval can be defined as follows:

$$[[BP]]\,\phi \equiv (\phi \wedge \pi) \vee \langle B\rangle\,(\phi \wedge \pi) \text{ and } [[EP]]\,\phi \equiv (\phi \wedge \pi) \vee \langle E\rangle\,(\phi \wedge \pi).$$

It should be noted here that in [Ven90] it was shown the modalities $\langle A\rangle$ and $\langle\bar{A}\rangle$ can be defined in terms other modal operators in the non-strict semantics (that is, partial ordering is non-strict):

$$\langle A\rangle\,\phi \equiv [[EP]]\left(\langle\bar{B}\rangle\,\phi\right) \text{ and } \langle\bar{A}\rangle\,\phi \equiv [[BP]]\left(\langle\bar{E}\rangle\,\phi\right)$$

Let be $\mathcal{M}$ be a model, $\mathcal{V} : \Phi \mapsto 2^{\mathbb{I}(\mathbb{T})}$, and $[t_1, t_2] \in \mathbb{I}(\mathbb{T})$. Assume that the partial ordering has a linear interval property. The formal semantics of HS formulas is then defined as follows:

$\mathcal{M}, [t_1, t_2] \models p$ iff $[t_1, t_2] \in \mathcal{V}\,(p)$ *for* $p \in \Phi$

$\mathcal{M}, [t_1, t_2] \models \neg\phi$ iff not $\mathcal{M}, [t_1, t_2] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \phi \wedge \psi$ iff $\mathcal{M}, [t_1, t_2] \models \phi$and $\mathcal{M}, [t_1, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \langle A\rangle\,\phi$ iff $(\exists t_3 : t_2 < t_3)$ s.t. $\mathcal{M}, [t_2, t_3] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \langle B\rangle\,\phi$ iff $(\exists t_3 : t_1 \leq t_3 < t_2)$ s.t. $\mathcal{M}, [t_1, t_3] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \langle E\rangle\,\phi$ iff $(\exists t_3 : t_1 < t_3 \leq t_2)$ s.t. $\mathcal{M}, [t_3, t_2] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \langle\bar{A}\rangle\,\phi$ iff $(\exists t_3 : t_3 < t_1)$ s.t. $\mathcal{M}, [t_3, t_1] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \langle\bar{B}\rangle\,\phi$ iff $(\exists t_3 : t_2 < t_3)$ $\mathcal{M}, [t_1, t_3] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \langle\bar{E}\rangle\,\phi$ iff $(\exists t_3 : t_3 < t_1)$ $\mathcal{M}, [t_3, t_2] \models \phi$

HS has enough expressive power to distinguish the different situations of time's being discrete, continuous, bound, linear or complete:

`length1` $\equiv\langle B\rangle\top \wedge [B]\,\pi$

`dense` $\equiv\neg$ `length1`

`discrete` $\equiv\pi \vee$ `length1` $\vee$ ($\langle B\rangle$`length1` $\wedge\langle E\rangle$`lenght1`)

`unbound` $\equiv \langle A\rangle\top\wedge \langle\bar{A}\rangle\top$

$\texttt{linear} \equiv (\langle A\rangle\,\phi \rightarrow [A](\phi\vee\,\langle B\rangle\,\phi\vee\,\langle \bar{B}\rangle\,\phi))\,\wedge$

$(\langle \bar{A}\rangle\,\phi \rightarrow [\bar{A}](\phi\vee\,\langle E\rangle\,\phi\vee\,\langle \bar{E}\rangle\,\phi))$ (where ϕ is a proposition)

$\texttt{complete} \equiv (\langle B\rangle\texttt{cell} \wedge [[EP]]\,\neg\phi \wedge [E]([[BP]]\,\phi\rightarrow\langle B\rangle\texttt{cell})) \rightarrow$

$\langle B\rangle([E](\neg\pi\rightarrow\langle D\rangle\texttt{cell}))$

(where $\texttt{cell} \equiv [[BP]]\,\phi\wedge\,[[EP]]\,\phi\wedge\,[D]\,\phi\,\wedge\,\langle D\rangle\,\phi$ and $\langle D\rangle\,\phi \equiv \langle B\rangle\,\langle E\rangle\,\phi$)

`length1` is true over an interval $[t_1, t_2]$ iff $t_1 < t_2$,and there are no points between t_1 and t_2. A temporal structure is discrete if the formula `discrete` is valid in that structure. Therefore, `discrete` is not satisfiable in $\mathbb{Q}$ or $\mathbb{R}$, but it is valid in $\mathbb{N}$. The formula `dense` is valid in $\mathbb{R}$ and $\mathbb{Q}$, but not in $\mathbb{N}$. `linear` is valid in linear temporal structures. The formula `complete` is valid in complete temporal structures. In particular, it is valid in $\mathbb{R}$, but not in $\mathbb{Q}$. This property distinguishes $\mathbb{R}$ from $\mathbb{Q}$, which are elementarily equivalent in first-order logic.

HS is a quite expressive logic due to its large modal operator set. However, it is not axiomatizable and highly undecidable [HS91]. The following theorems are taken from [HS91].

The validity problem for any class of temporal structures that contains an infinitely ascending sequence is r.e.-hard.

HS is undecidable for the class of all models, linear models, discrete models, dense models, and dense, linear, unbounded models.

The validity problem for complete classes of temporal structures which contain an infinitely ascending sequence is $\prod_1^1$-hard.

HS is undecidable for the orderings of the natural numbers, integers, or reals.

Undecidability even occurs in the classes of structures with no infinitely ascending sequences.

The validity problem for any complete class of temporal structures which has unboundedly ascending sequences is co-r.e.-hard.

Undecidability results given above can be proved using an infinitely ascending sequence in the model to simulate the halting problem for Turing machines. Any unbounded ordering contains an infinite ascending sequence. A class of ordered structures contains an infinite ascending sequence if at least one structure in the class includes an infinite ascending sequence [GMS04]. In [MR99] undecidability was proved by means of tiling problem.

In [Ven90] some interesting results for the logic HS were presented. By using a geometrical representation for the modalities Venema introduced a sound and complete proof system for HS. He has also proved that HS is strictly more expressive than any temporal logic based on linear orderings of time instants.

In [HS91] a translation machinery that converts a HS formula to its equivalent first-order formula on a corresponding first-order structure was provided. Such as translation is useful to reduce problems to well-known results in first-order logic.

6.2.2 The Logic BE

The logic BE is a fragment of HS containing the modal operators $\langle B \rangle$ and $\langle E \rangle$. The syntax can be recursively defined as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle B \rangle \, \phi \mid \langle E \rangle \, \phi$$

BE can express the conditions on the underlying interval structure as with HS, except the formula `linear`. In the literature there is no result concerning the decidability of the satisfiability problem over strict models, or a sound and complete proof system for BE. In [Lod00] the satisfiability problem for BE formulas interpreted over all non-strict linear structures was found to be undecidable. The proof is based on reducing the non-halting problem of a Turing machine on a blank tape to the satisfiability problem for BE.

6.2.3 *The Logic D*

The logic is a sub-interval relation, which only allows to look inside the current interval [HS91]. The formulas of the logic D are recursively defined as follows:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle D \rangle \, \phi$$

Besides the strict and non-strict versions, the logic D has different semantic variations, depending on which sub-interval relation is assumed. Thus, the semantic definition of modal operator $\langle D \rangle$ is defined as follows:

$\mathcal{M}, [t_1, t_2] \models \langle D \rangle \, \phi$ iff there exists an sub-interval $[t'_1, t'_2] of \; [t_1, t_2]$ such that $\mathcal{M}, [t'_1, t'_2] \models \phi$

The sub-interval logic was first studied in [vB91], where the subinterval relation $\subseteq$ is considered. When the strict semantics is considered and formulas are interpreted over the rational numbers or the class of all linear orderings, the logic D becomes equivalent to the standard modal logic S4. When formulas are interpreted over integers, the logic D becomes equivalent to the modal logic S4 with *Grzegorczyk's Axiom,* $[D] \, ([D](p \rightarrow [D]p) \rightarrow p) \rightarrow p$, expressing that $\subseteq$ is well-formed. The satisfiability problem for both S4 and S4 with *Grzegorczyk's Axiom* is known to be PSPACE-complete ([CR03], [DG00]).

In the literature there is no information about decidability results and axiomatization systems for the logic D when the non-strict semantics and/or the proper non-strict subintreval relation $\subset$ is considered over any class of linear orders.

6.2.4 *Propositional Neighbourhood Logic*

Propositional neighbourhood logic is the propositional fragment of first-order neighbourhood logics introduced in [CH98]. It has been studied on both strict and non-strict linear structures [GMS03a]. The language with non-strict semantics is called $\mathrm{PNL}^{\pi+}$, and it includes the modal operators $\Diamond_r$ (*met by*) and $\Diamond_l$ (*meets*), and the model constant π. The formulas of $\mathrm{PNL}^{\pi+}$ is defined as follows:

$$\phi ::= p \mid \pi \mid \neg\phi \mid \phi \wedge \psi \mid \Diamond_r\phi \mid \Diamond_l\phi$$

The dual operators, $\Box_l$ and $\Box_r$, can be defined as usual.

Let be $\mathcal{M}$ be a model, $\mathcal{V} : \Phi \mapsto 2^{\mathbb{I}(\mathbb{T})}$, and $[t_1, t_2] \in \mathbb{I}(\mathbb{T})$. The formal semantics of $\mathrm{PNL}^{\pi+}$ formulas is then defined as follows:

$\mathcal{M}, [t_1, t_2] \models p$ iff $[t_1, t_2] \in \mathcal{V}(p)$ *for* $p \in \Phi$

$\mathcal{M}, [t_1, t_2] \models \pi$ iff $t_1 = t_2$

$\mathcal{M}, [t_1, t_2] \models \neg\phi$ iff not $\mathcal{M}, [t_1, t_2] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \phi \wedge \psi$ iff $\mathcal{M}, [t_1, t_2] \models \phi$and $\mathcal{M}, [t_1, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \Diamond_r\phi$ iff $(\exists t_3 \geq t_2)$ s.t. $\mathcal{M}, [t_2, t_3] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \Diamond_l\phi$ iff $(\exists t_3 \leq t_1)$ s.t. $\mathcal{M}, [t_3, t_1] \models \phi$

Note that the semantic definition given above is for non-strict linear orderings. In case of strict linear structures semantics of the modal operators can be defined similarly:

$\mathcal{M}, [t_1, t_2] \models \Diamond_r\phi$ iff $(\exists t_3 > t_2)$ such that $\mathcal{M}, [t_2, t_3] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \Diamond_l\phi$ iff $(\exists t_3 < t_1)$ such that $\mathcal{M}, [t_3, t_1] \models \phi$

Assume PNL^{+} denotes the non-strict PNL without the modal constant π, and PNL^{-} denotes the strict PNL without the modal constant π . It is known that the logic $\mathrm{PNL}^{\pi+}$ subsumes both PNL^{+} and PNL^{-}.

Given that formulas are interpreted over strict linear models, PNL^{-} has enough expressive power to distinguish the different classes of linear structures, such as discreteness, continuity, boundness, or completeness:

`unbound` $\equiv \Box_r\phi \rightarrow \Diamond_r\phi$

`dense` $\equiv (\Diamond_r\Diamond_r\phi \rightarrow \Diamond_r\Diamond_r\Diamond_r\phi) \wedge (\Diamond_r\Box_r\phi \rightarrow \Diamond_r\Diamond_r\Box_r\phi)$

`discrete` $\equiv (\Box_r\bot \rightarrow \Box_l(\Box_r\Box_r\bot \vee \Diamond_r(\Diamond_r\top \wedge \Box_r\Box_r\bot))) \wedge$

$$((\Diamond_r\top \wedge \Box_r(\phi \wedge \Box_l\neg\phi \wedge \Box_r\phi)) \rightarrow \Box_l\Box_l\Diamond_r(\Diamond_r\neg\phi \wedge \Box_r\Box_r\phi))$$

$\texttt{complete} \equiv \Diamond_r \Diamond_r \Box_l \phi \land \Diamond_r \Box_r \neg \Box_l \phi \to \Diamond_r \left(\Diamond_r \Box_l \Box_l \phi \land \Box_r \Diamond_r \neg \Box_l \phi \right)$

Since PNL^- can be encoded into $\mathrm{PNL}^{\pi+}$, the above classes of structures can also be defined in $\mathrm{PNL}^{\pi+}$.

In [GMS03a] several sound and complete axiomatic systems are provided for various classes of models, including strict linear structures, strict linear structures, complete unbounded linear structures, unbounded structures, dense structures, discrete structures, dense unbounded structures and discrete unbounded structures.

As for decidability results, in [BMS07a] the authors show that the satisfiability problem for $\mathrm{PNL}^{\pi+}$, PNL^+ and PNL^-over the integers is NEXPTIME-complete. They develop a sound and complete tableau-based decision procedure, and prove that it is optimal. In [BGMS07], the expressive power of $\mathrm{PNL}^{\pi+}$, PNL^+ and PNL^- is compared, and it is shown that $\mathrm{PNL}^{\pi+}$ is strictly more expressive than PNL^+ and PNL^-. Then, the authors prove that the satisfiability problem for $\mathrm{PNL}^{\pi+}$over the class of all linear orders, as well as over some natural subclasses of it, such as the class of all well-orders and the class of all finite linear orders, can be decided in NEXPTIME by reducing it to the satisfiability problem for the two-variable fragment of first-order logic over the same classes of structures.

An important fragment of the propositional neighbourhood logic is the *right propositional neighbourhood logic* (RPNL) which is based on the right neghbourhood relation between intervals. As in the case of propositional neighbourhood logic, the language with non-strict semantics is called $\mathrm{RPNL}^{\pi+}$. The non-strict fragment without the modal constant π is denoted by RPNL^+, and the strict fragment without the modal constant π is denoted by RPNL^-. As for decidability results, in [BM05] an EXSPACE tableau-based decision procedure is devised for RPNL^- interpreted over natural numbers. In [BMS07b] the authors develop an alternative NEXPTIME decision procedure that works for all variants of RPNL ($\mathrm{RPNL}^{\pi+}$, RPNL^+, and RPNL^-) interpreted over natural numbers, and they prove its optimality.

6.2.5 *The Logic CDT*

The Logic CDT was introduced by Venema in [Ven91]. It is the most expressive propositional interval logic over non-strict linear structures. The formulas of CDT logic is recursively defined as follows:

$$\phi ::= p \mid \pi \mid \neg\phi \mid \phi \land \psi \mid \phi C \psi \mid \phi D \psi \mid \phi T \psi$$

Assume that the total ordering is non-strict. The formal semantics of CDT formulas is then defined as follows:

$\mathcal{M}, [t_1, t_2] \models \pi$ iff $t_1 = t_2$

$\mathcal{M}, [t_1, t_2] \models p$ iff $[t_1, t_2] \in \mathcal{V}(p)$ *for* $p \in \Phi$

$\mathcal{M}, [t_1, t_2] \models \neg\phi$ iff not $\mathcal{M}, [t_1, t_2] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \phi \wedge \psi$ iff $\mathcal{M}, [t_1, t_2] \models \phi$and $\mathcal{M}, [t_1, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \phi C \psi$ iff $(\exists t_3 : t_1 \leq t_3 \leq t_2)$ s.t. $\mathcal{M}, [t_1, t_3] \models \phi$ and $\mathcal{M}, [t_3, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \phi D \psi$ iff $(\exists t_3 : t_3 \leq t_1)$ s.t. $\mathcal{M}, [t_3, t_1] \models \phi$ and $\mathcal{M}, [t_3, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \phi T \psi$ iff $(\exists t_3 : t_2 \leq t_3)$ s.t. $\mathcal{M}, [t_2, t_3] \models \phi$ and $\mathcal{M}, [t_1, t_3] \models \psi$

Since the logic HS is the propositional interval logic of Allen's relations, every propositional interval logic with unary modalities based on Allen's relations is subsumed by CDT:

$\langle B \rangle \varphi = \varphi C(\neg\pi)$; $\langle \overline{B} \rangle$=$(\neg\pi)T\varphi$;

$\langle E \rangle \varphi = (\neg\pi)C\varphi$; $\langle \overline{E} \rangle$=$\varphi D(\neg\pi)$;

$\langle A \rangle \varphi = (\neg\pi \wedge \varphi)T\top$; $\langle \overline{A} \rangle$=$(\neg\pi \wedge \varphi)D\top$;

In [Ven91] a sound and complete axiomatic system is provided for the logic CDT which is interpreted over non-strict linear models. This axiomatic system can be extended for the classes of discrete linear orderings, dense linear orderings, etc. As a consequence of the previous results for the logic HS, the satisfiability problem for CDT is not decidable over almost all interesting classes of linear ordering, including $\mathbb{N}$,$\mathbb{Z}$,$\mathbb{R}$, etc.

A generalization of CDT to (non-strict) partial orderings with the linear interval property, called BCDT$^+$, has been recently introduced in [GMS03b]. BCDT$^+$ features the same operators as CDT; however, it is interpreted over partially ordered domains with linear intervals. The decidability and axiomatizability of the strict versions of CDT and BCDT$^+$ have not been studied yet; but it is natural to expect that similar results aply there too.

6.2.6 *The Logic PITL*

PITL is the propositional fragment of the First-order Interval Temporal Logic (ITL) (see Section 3.2). The formulas of PITL logic is recursively defined as follows:

$$\phi ::= p \mid \pi \mid \neg\phi \mid \phi \wedge \psi \mid \phi C$$

The modalities $\langle B \rangle$ and $\langle E \rangle$ can be defined as follows:

$$\langle B \rangle \phi \equiv \phi C \pi \text{ and } \langle E \rangle \phi \equiv \neg\pi C$$

PITL was originally restricted to the class of discrete linear orderings with finite time, with the chop operator is paired with a next operator $\bigcirc$, instead of π. Intervals in such structures will be identified with the finite sequences of points.

The satisfiability problem for PITL interpreted over the class of non-strict discrete structures is undecidable [Mos83].

The satisfiability problem for PITL is reduced to checking the emptiness of the intersection of two context-free grammars. A PITL formula is satisfiable if and only if the intersection of the languages generated by two context-free grammars, over which PITL formula is constructed, is not empty. This problem is known to be undecidable; thus the satisfiability problem for PITL is undecidable.

The satisfiability problem for PITL interpreted over non-strict linear structures and non-strict dense linear structures is also undecidable.

Decidable variants of PITL, interpreted over discrete models, can be obtained by imposing the *locality projection principle*. Locality projection means that each propositional variable true over an interval if and only if it is true at its first state. This allows one to collapse all the intervals starting at the same state into the single interval consisting of the first state only. Let LPITL be the logic obtained by imposing the locality projection principle to PITL. The syntax of LPITL coincides with that of PITL. However, in LPITL propositional variables evaluated over points instead of intervals.

[Mos83] introduced the logic QLPITL, which is an extension of LPITL over finite time with quantification over propositional variables. It was also shown that the satisfiability problem for the loic QLPITL interpreted over the class of non-strict discrete linear structures is (non-elementarily) decidable.

LPITL was also extended with the *chop-star* modality, denoted by * [Mos83, Mos00a, Mos00b, Mos03]. For any ϕ, ϕ^* holds over a given discrete interval if and only if the interval can be chopped into zero or more parts such that ϕ holds over each of them. The resulting logic is called LPITL*, and it is interpreted over finite or infinite discrete linear structures.

In [Mos03] a sound and complete axiomatic system is provided for LPITL* which is interpreted over non-strict discrete linear structures.

6.3 First-Order Interval Temporal Logics

First-order interval temporal logics have been introduced as a tool for the formal specification and verification of hardware real time systems. ITL is the most commonly known first-order interval temporal logic. Numerous extensions of ITL have been shown to be useful in the specification of various kinds of software and hardware systems.

6.3.1 *The Logic ITL*

ITL, interpreted over discrete linear orderings with finite time intervals, was first introduced in [Mos83]. The formulas of ITL are constructed from the following: an infinite set of global (independent of time and time intervals) variables $x, y, z, ...$, an infinite set of temporal variables $t, t', ...$, an infinite set of global function symbols $f^n, g^m, ...$, where f^n is a function of arity n and g^m is a function of arity m, an infinite set of predicate symbols $P^n, R^m, ...$, where P^n is a predicate of arity n and R^m is a predicate of arity m, an infinite set of temporal propositional letters $X, Y, ...$. The set of terms θ, θ_i is defined by the following abstract syntax:

$$\theta ::= x \mid t \mid f^n(\theta_1, ..., \theta_n)$$

The formulas of ITL can be recursively defined as follows:

$$\phi ::= X \mid P^n(\theta_1, ..., \theta_n) \mid \neg\phi \mid \phi \wedge \psi \mid \phi C \psi \mid (\exists x)\,\phi$$

Let Δ be the set of temporal variables, Φ be the set of temporal propositional letters and $\mathcal{I}$ be the set of all bounded and closed intervals of real numbers $\{[t_1, t_2] : t_1 \leq t_2 \wedge t_1, t_2 \in \mathbb{R}\}$. The meanings of temporal variables and propositional letters, i.e. the interval-dependent symbols, are given by the interpretation:

$$\mathcal{J} \in (\Delta \to (\mathcal{I} \to \mathbb{R})) \cup (\Phi \to (\mathcal{I} \to \{true, false\}))$$

where $\mathcal{J}(t)([t_1, t_2]) \in \mathbb{R}$ for all $t \in \Delta$, $\mathcal{J}(\ell)([t_1, t_2]) = t_2 - t_1$ (ℓ is a special temporal variable denoting the interval length) , $\mathcal{J}(X)([t_1, t_2]) \in \{true, false\}$ for all $X \in \Phi$.

A valuation is a mapping $\mathcal{V}$ which associates a real number with each global variable. Given a variable x, two valuations $\mathcal{V}$ and $\mathcal{V}'$ are said to be $x - equivalent$ if $\mathcal{V}(y) = \mathcal{V}'(y)$ for every global variable y which is different from x.

Assume that a total function $\underline{f^n} \in \mathbb{R}^n \to \mathbb{R}$ is associated with each n-ary function symbol f^n. The semantics of a term θ at an interval $[t_1, t_2]$ under a valuation $\mathcal{V}$ is denoted by $\mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(\theta)$. The function $\mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}$ is recursively defined as follows:

- for a global variable x, $\mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(x) = \mathcal{V}(x)$

- for a temporal variable t, $\mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(t) = \mathcal{J}(t)([t_1, t_2])$

- for a term θ of the form $f^n(\theta_1, ..., \theta_n)$, $\mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(\theta) = \underline{f^n}(\alpha_1, ..., \alpha_n)$ where $\alpha_i = \mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(\theta_i)$ for $1 \leq i \leq n$

Assume that a total function $\underline{G^n} \in \mathbb{R}^n \rightarrow \{true, false\}$ is associated with each n-ary relation symbol G^n. Let $\mathcal{M} = \langle \mathcal{J}, \mathcal{V} \rangle$ be a model for ITL. The formal semantics of ITL formulas is then defined as follows:

$\mathcal{M}, [t_1, t_2] \models X$ iff $\mathcal{J}(X)([t_1, t_2]) = true$

$\mathcal{M}, [t_1, t_2] \models G^n(\theta_1, ..., \theta_n)$ iff $\underline{G^n}(\alpha_1, ..., \alpha_n) = true$ where $\alpha_i = \mathcal{J}^{\mathcal{V}}_{[t_1,t_2]}(\theta_i)$ for $1 \leq i \leq n$

$\mathcal{M}, [t_1, t_2] \models \neg\phi$ iff not $\mathcal{M}, [t_1, t_2] \models \phi$

$\mathcal{M}, [t_1, t_2] \models \phi \wedge \psi$ iff $\mathcal{M}, [t_1, t_2] \models \phi$ and $\mathcal{M}, [t_1, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models \phi C \psi$ iff $(\exists t_3 : t_1 \leq t_3 \leq t_2)$ $\mathcal{M}, [t_1, t_3] \models \phi$ and $\mathcal{M}, [t_3, t_2] \models \psi$

$\mathcal{M}, [t_1, t_2] \models (\exists x)\, \phi$ iff $\mathcal{M}, [t_1, t_2] \models \phi$ for some value assignment $\mathcal{V}'$ which is x-equivalent to $\mathcal{V}$

A sound and complete axiomatic system is represented in [Dut95]. A term or formula is called *flexible* if a temporal variable including the symbol ℓ or a propositional letter occurs in the term or formula. A term or formula which is not flexible is called *rigid*.

Sound and complete axiomatic systems for local variants of ITL (with locality constraint) for finite and infinite time were presented in [Dut95, Mos00b]. ITL was extended with projection in [Gue00a] where a complete axiomatic system is provided. A probabilistic extension of ITL was studied in [Gue00b]. Not surprisingly ITL is highly undecidable.

6.3.2 *The Logic NL*

ITL does not allow looking outside of the current interval. The logic NL was proposed in [CHR91], where the left *neighbourhood modality* $\Diamond_l$ and right *neighbourhood modality* $\Diamond_r$ were introduced. The formulas of NL can be recursively defined as follows:

$$\phi ::= X \mid P^n(\theta_1, ..., \theta_n) \mid \neg\phi \mid \phi \wedge \psi \mid \Diamond_l \phi \mid \Diamond_r \phi \mid (\exists x)\, \phi$$

where X and θ_i's are defined as in ITL. The semantics of the modal operators $\Diamond_l$and $\Diamond_r$ is defined as in PNL, and the rest of the semantics is defined as in ITL. NL can express any of the modalities corresponding to the Allen's relations; thus, it can represent the properties of the underlying linear ordering, such as discreteness, density, etc. For example, the chop operator C can be expressed in terms of the modalities $\Diamond_l$ and $\Diamond_r$ as follows:

$$\phi C \psi = \exists x, y\, (\ell = x + y) \wedge \Diamond_l \Diamond_r ((\ell = x) \wedge \phi \wedge \Diamond_r ((\ell = y) \wedge \psi))$$

In [BC97] *up* and *down modalities,* represented by $\Diamond_u$, $\Diamond_d$ respectively, were introduced, and two dimensional version of NL, called NL^2, was proposed. In [BRC00] a sound and complete axiomatic system is provided for the logic NL. NL is an undecidable logic as in ITL.

6.3.3 *Duration Calculus*

Duration Calculus (DC) is a first-order interval temporal logic with the additional notion of *state*, which is characterized by a *duration*. The duration of a state is the length of the time period during which the system remains in the state. Duration calculus has been successfully applied to the specification and verification of real-time systems [GMS04]. It was first introduced in [CHR91], where formulas are interpreted over the class of non-strict interval structures based on $\mathbb{R}$.

The formulas of DC can be recursively defined as follows:

$$\phi ::= P^n(\theta_1, ..., \theta_n) \mid \neg\phi \mid \phi \vee \psi \mid \phi C \psi \mid (\exists x)\,\phi$$

where θ_i's are terms as defined in ITL, P^n is an n-ary predicate, C is the chop modality, and x is a global variable.

Duration calculus is an extension of ITL in the sense that temporal variables other than ℓ have a structure $\int S$, where $\int S$ is called a *state duration* and S is called a *state expression*. The set of state expressions is generated from a set of *state variables* $P, Q, R, ...$, according to following abstract syntax:

$$S ::= 0 \mid 1 \mid P \mid \neg S_1 \mid S_1 \vee S_2$$

Let $\mathcal{S}$ be a set of state of state variables, Φ be the set of temporal propositional letters and $\mathcal{I}$ be the set of all bounded and closed intervals of real numbers $\{[t_1, t_2] : t_1 \leq t_2 \wedge t_1, t_2 \in \mathbb{R}\}$. The meanings of state variables, temporal variable ℓ, and propositional letters are given by the interpretation:

$$\mathcal{J} \in (\mathcal{S} \rightarrow (\mathbb{R} \rightarrow \{0,1\})) \cup (\{\ell\} \rightarrow (\mathcal{I} \rightarrow \mathbb{R})) \cup (\Phi \rightarrow (\mathcal{I} \rightarrow \{true, false\}))$$

where $\mathcal{J}(S)(t) \in \{0,1\}$ for all state variables $S \in \mathcal{S}$ and $t \in \mathbb{R}$, $\mathcal{J}(\ell)([t_1, t_2]) = t_2 - t_1$, $\mathcal{J}(X)([t_1, t_2]) \in \{true, false\}$ for all $X \in \Phi$.

Given the interpretation $\mathcal{J}$, the semantics of a state expression S is a total function $\Im[S]$: $\mathbb{R} \rightarrow \{0,1\}$ which has a finite number of discontinuity points only. For any time point t, the semantics can be defined inductively on the structure of state expressions as follows:

$\Im[0](t) = 0$

$\Im[1](t) = 1$

$\Im[P](t) = \mathcal{J}(P)(t)$

$\Im[\neg S_1](t) = 1 - \Im[S_1](t)$

$\Im[S_1 \vee S_2](t) = \begin{cases} 0\ if\ \Im[S_1](t) = 0\ and\ \Im[S_2](t) = 0 \\ 1\ otherwise \end{cases}$

The semantics of a duration $\int S$ in a given model, with respect to an interval $[t_1, t_2]$, can be defined by $\Im\left[\int S\right]([t_1, t_2]) = \int_{t_1}^{t_2} \Im[S](t)\, dt$.

For a given two interpretations $\Im$ and $\Im'$whose values for any state variable S disagree in at most a finite number of points in any interval we have

$$\Im\left[\int S\right]([t_1, t_2]) = \Im'\left[\int S\right]([t_1, t_2])$$

One can define some useful abbreviations in DC:

$\lceil S \rceil \equiv \ell = 0$

$\lceil S \rceil \equiv \int S = \ell \wedge \ell > 0$

Here $\lceil S \rceil$ stands for : "S holds almost everywhere over a strict interval". $\int 1$, usually abbreviated by ℓ, can be viewed as the length of the current interval.

All axioms and inference rules of ITL can be adopted for DC. However, additional axioms are needed for temporal variables.

In [CH04] a sound and complete axiomatic system is provided for Duration Calculus. The satisfiability problem for both first-order and propositional DC has been shown to be undecidable [CHS93]. Various fragments of DC have been investigated so far. In [CHS93] a fragment of propositional DC, called RDC, was introduced. RDC formulas are generated by

if S is a state expression, then $\lceil S \rceil \in$ RDC

if $\phi, \psi \in RDC$, then $\neg\phi$, $\phi \vee \psi, \phi C \psi \in$ RDC

It was shown that RDC has a decidable satisfiability problem when interpreted over $\mathbb{N}$, $\mathbb{Q}$ and $\mathbb{R}$. In [Rab98] it was shown that the satisfiability problems for RDC interpreted over $\mathbb{N}$ and RDC interpreted over $\mathbb{R}$ are decidable by providing a linear time reduction from the equivalence problem for star-free expressions to the validity problem for the considered fragment of DC. When RDC is extended with the formulas of the type $\ell = k$, the satisfiability problem remains decidable over $\mathbb{N}$. However, the problem becomes undecidable over

$\mathbb{Q}$ and $\mathbb{R}$. The fragment of propositional DC whose formulas are constructed from primitive formulas of the type $\int S_1 = \int S_2$ is undecidable. In [Fra96] the decidable class of RDC was extended for continuous time by including $\int S = k$, but with a restriction on the finite variability such that the number of discontinuous points of any state in any unit interval has a fixed upper bound. In [Gra99] a decidability result was presented for a variant of DC where negation is removed from RDC but an iteration operator is introduced together with the inequalities $\ell \geq k$ and $\ell \leq k$, where $k \in \mathbb{N}$. In [CSDC00] another fragment of propositional DC was introduced by imposing some syntactic restrictions. It is shown that this logic is expressive enough to capture Allen's relations. By proposing a sound, complete and terminating tableau system for the logic it is shown that the satisfiability problem is decidable. The tableau system is a mixed procedure, combining standard tableau techniques with temporal constraint network resolution algorithms. In [Pan01] quantification over states are introduced. The satisfiability of formulas is still decidable. This decision algorithm was implemented as a tool called DCVALID. In [CH98] Duration Calculus and first-order neighbourhood logic were combined, and a axiomatic systems for DC and NL were merged. The fragment of DC/NL obtained by restricting the formulas to be constructed only form the primitive formulas of the type $\lceil S \rceil$ was proved to be decidable, while extension of the extension with the formulas of the type $\ell = k$ is undecidable.

DC has been applied to several areas, such as real-time and hybrid systems. Automatic verification and model-checking tools have been developed and analyzed. In [Fra96] model checking methods for DC were described, and it was argued that the class of models is restricted to the possible behaviours of embedded real-time systems, model checking procedures are feasible for rich subsets of DC.

6.4 Temporal Logics for Events and States

Researchers have tried to develop computationally manageable logics which can capture the formal semantics of temporal constructions in natural language [Dow79, CP93, HS94, Ter96]. However, these formal systems exhibit high computational complexity. In general, semantics of temporal constructions are represented in a first-order interval temporal logic where variables range over time-intervals, predicates correspond to event-types and temporal-order relations. Such a logic is undoubtedly undecidable. Using temporal logics with limited expressive power is motivated by work in natural language semantics and discourse. TPL, introduced in [PH05], is such an interval temporal logic with a limited expressive power. A simple context-free grammar is used to define a fragment of English featuring the temporal constructions. The phrase-structures assigned to this fragment can be viewed as expressions in TPL. In [PH05] it has been shown that the satisfiability problem is NEXPTIME-complete.

In the sequel, $\mathcal{I}$ denotes the set of intervals, where an interval is closed, bounded, non-empty subset of $\mathbb{R}$. Temporal variables are denoted by the variables $I, J, ...$, which range over $\mathcal{I}$. Given that I represents the interval $[t_1, t_2]$ and J represents the interval $[t_3, t_4]$ where $t_1, t_2, t_3, t_4 \in \mathbb{R}$ and $t_1 \leq t_3 \leq t_4 \leq t_2$, the partial functions $init\,(J, I)$ and $fin\,(J, I)$ denote the intervals $[t_1, t_3]$ and $[t_4, t_2]$, respectively. $J \subset I$ means that J is a strict subset of I. Similarly, $J \subseteq I$ means that J is a non-strict subset of I.

Assume that $\mathcal{E}$ is a fixed infinite set of event atoms, and $e \in \mathcal{E}$. The set of event relations α is defined by the following abstract syntax:

$$\alpha ::= e \mid e^f \mid e^l$$

where the letter f and l stand for the adjectives *first* and *last,* respectively. That is, e^f denotes the first of finitely many events of type e, and e^l denotes the last of finitely many events of type e within a temporal context I. However, in some cases finding the first or last of events might be ambiguous. This might happen when an event e does not begin or end before all others. In [PH05] this ambiguity is resolved as follows: Let $\mathcal{J}$ be the non-empty set of all proper subintervals of I over which an event e occurs, and $\mathcal{J}'$ be the non-empty subset whose elements have the earliest end-point. If we select $J \in \mathcal{J}'$ whose start-point is latest, we get the smallest of the proper subintervals which have the earliest end-point, and over which the event e occurs. Thus, the phrase *the first* e denotes the interval J within a temporal context I. Similiarly, the phrase *the last* e denotes the smallest of the proper subintervals which have the latest beginning-point, and over which the event e occurs.

The formulas of TPL can be recursively defined as follows:

$$\phi ::= \top \mid \neg\phi \mid \langle e \rangle\, \phi \mid [e]\, \phi \mid \{\alpha\}\, \phi \mid \{\alpha\}_{>}\, \phi \mid \{\alpha\}_{<}\, \phi \mid \phi \wedge \psi \mid \phi \vee \psi$$

The connectives $\rightarrow$ and $\leftrightarrow$ can be defined in usual way.

Let $\mathcal{E}$ be a fixed infinite set of events atoms, and $\mathcal{I}$ be the set of all bounded, closed, non-empty intervals of real numbers $\{[t_1, t_2] : t_1 \leq t_2 \wedge t_1, t_2 \in \mathbb{R}\}$. A 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 $\mathcal{M}\,(e)$ can be defined as follows:

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

where $\langle J, e \rangle$ represents that the event e occurs over the interval J. Here we assumed that models be finite. This is due to the fact that there are situations in which event atoms are instantiated in finite contexts. The formulas which require infinite models are not satisfiable in TPL. For example, $\langle e \rangle \top \wedge [e] \langle e \rangle \top$is unsatisfiable since the formula requires an infinite model.

Let $\mathcal{M}$ be a model, and $I, J \in \mathcal{I}$. The semantics of the event-relation α can be defined as follows:

$\mathcal{M}, J \models_I e$ iff $J \subset I$ and $e \in \mathcal{M}(J)$

$\mathcal{M}, J \models_I e^f$ iff $\mathcal{M}, J \models_I e$ and J is the minimal-first interval of I

$\mathcal{M}, J \models_I e^l$ iff $\mathcal{M}, J \models_I e$ and J is the minimal-last interval of I

Minimal- and maximal- first interval can be defined by the following definition: Assume that $J = [t_1, t_2] \subset I$ where J satisfies some property $\mathcal{P}$. J is the *minimal-first subinterval* of I if for every $J' = [t'_1, t'_2] \subset I$ where J satisfies the property $\mathcal{P}$ either $t_2 < t'_2$ or $t_2 = t'_2$ and $t_1 \geq t'_1$. Similarly, J is the *minimal-last subinterval* of I if for every $J' = [t'_1, t'_2] \subset I$ where J satisfies the property $\mathcal{P}$ either $t_1 > t'_1$ or $t_1 = t'_1$ and $t_2 \leq t'_2$. Since $\mathcal{M}$ is finite if there exists any $J \subset I$ with $\langle J, e\rangle \in \mathcal{M}$, then the minimal-first and minimal-last interval J exist, and they are unique.

Let $\mathcal{M} = \langle \mathcal{I}, \mathcal{E}\rangle$ be a model for TPL and $I \in \mathcal{I}$. The formal semantics of TPL formulas is then defined as follows:

$\mathcal{M} \models_I \top$is always true

$\mathcal{M} \models_I \langle e\rangle\, \phi$ iff there is a $J \subset I$ such that $\mathcal{M}, J \models_I e$ and $\mathcal{M} \models_J \phi$

$\mathcal{M} \models_I [e]\, \phi$ iff $\mathcal{M}, J \models_I e$ implies $\mathcal{M} \models_J \phi$ for all $J \subset I$

$\mathcal{M} \models_I \{\alpha\}\, \phi$ iff there is a unique $J \subset I$ such that $\mathcal{M}, J \models_I \alpha$ and $\mathcal{M} \models_J \phi$

$\mathcal{M} \models_I \{\alpha\}_<\, \phi$ iff there is a unique $J \subset I$ such that $\mathcal{M}, J \models_I \alpha$ and $\mathcal{M} \models_{init(J,I)} \phi$

$\mathcal{M} \models_I \{\alpha\}_>\, \phi$ iff there is a unique $J \subset I$ such that $\mathcal{M}, J \models_I \alpha$ and $\mathcal{M} \models_{fin(J,I)} \phi$

$\mathcal{M} \models_I \neg\phi$ iff not $\mathcal{M} \models_I \phi$

$\mathcal{M} \models_I \phi \wedge \psi$ iff $\mathcal{M} \models_I \phi$ and $\mathcal{M} \models_I \psi$

$\mathcal{M} \models_I \phi \vee \psi$ iff $\mathcal{M} \models_I \phi$ or $\mathcal{M} \models_I \psi$

In [PH05] it is shown that the satisfiability problem is NEXPTIME-complete. Upper complexity bound is achieved by establishing an exponential bound on the size of satisfying models. This is done by constituting a terminating tableu decision procedure for TPL, interpreted over a linear time flow, and finite models including finitely many events that occur over bounded intervals. The hardness-proof is done by encoding any instance of any exponential tiling problem as an instance of the satisfiability problem of TPL.

In [Kon06] TPL is extended with the notion of state models and durations, and the new logic is called TPL*. Thus, TPL* contains both event-based and state-based views, and it is more suitable than TPL for specifying properties of finite sequences of states, and presenting the semantics of temporal constructions in English. TPL* also includes the *chop modality* C which is necessary to capture important real-time problems like behaviour of complex systems. [Kon06] also proposes a terminating tableu system for the logic TPL*, thus showing that its satisfiability problem is decidable. Indeed, this problem is still NEXTIME-complete.

7 Conclusion

In this survey paper we have outlined recent important developments on temporal logics by presenting various formal systems dealing with various time structures, and discussing important features, such as (un)decidability results, expressiveness and axiomatization systems.

References

[1] [Aba89] Abadi M. The power of temporal proofs. *Theoretical Computer Science*, vol. 64, pp. 35-84, 1989.

[2] [Abr80] Abrahamson K. *Decidability and expressiveness of logics of programs*. Ph.D. Thesis, University of Washington, 1980.

[3] [AF94] Allen J. F. and Ferguson G. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation*, vol. 4, num. 5, pp. 531-579, 1994.

[4] [AH85] Allen J. F. and Hayes J. P. A Common-Sense Theory of Time. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 528-531, 1985.

[5] [AH89] Allen J. F. and Hayes J. P. Moments and Points in an Interval-based Temporal Logic. *Computational Intelligence* 5, pp. 225-238, 1989.

[6] [All83] Allen J. F. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, vol. 26, pp. 832-843, 1983.

[7] [All84] Allen J. F. Towards a General Theory of Action and Time. *Artificial Intelligence* vol. 23, pp. 123-154, 1984.

[8] [ANS79] Andreka H., Nemeti I. and Sain I. Completeness problems in verification of programs and program schemes. In *Mathematical Foundations of Computer Science*, *Lecture Notes in Computer Science*, Springer, 1979.

[9] [ANvB98] Andreka H., Nemeti I. and van Benthem J. Modal languages and bounded fragments of predicate logic. J*ournal of Philosophical Logic*, vol. 27, pp. 217-274, 1998.

[10] [AVFY98] Abiteboul S., Vianu V., Fordham B. and Yesha Y. Relational transducers for electronic commerce. In *PODS'98*, pp. 179-187, ACM Press, 1998.

[11] [BC97] Barua R., Chaochen Z. Neighbourhood Logics. *NL and NL2*, report num. 120, UNU/IIST, 1997.

[12] [BGG97] Boerger E., Graedel E. and Gurevich Y. The Classical Decision Problem. *Perspectives in Mathematical Logic*, Springer, 1997.

[13] [BGMS07] Bresolin D., Goranko V., Montanari A., and Sciavicco G. On Decidability and Expressiveness of Propositional Interval Neighbourhood Logics. In *Proceedings of the International Symposium on Logical Foundations of Computer Science (LFCS)*, vol. 4514 of LNCS, pp. 84-99, 2007.

[14] [BKP86] Barringer H., Kuiper R. and Pnueli A. A Really Abstract Concurrent Model and its Temporal Logic. *POPL*, pp. 173-183, 1986.

[15] [BM05] Bresolin D. and Montanari A. A Tableau-based Decision Procedure for Right Propositional Neighbourhood Logic. In *Proceedings of TABLEAUX 2005*, vol. 3702 of LNAI, pp. 63-77, 2005.

[16] [BMP81] Ben-Ari M., Manna Z. and Pnueli A. The temporal logic of branching time. In *Proceedings of the 8th Annual ACM Symposium on Principles of Programming Languages*, pp. 164-176, 1981.

[17] [BMS07a] Bresolin D., Montanari A. and Sala P. An Optimal Tableau-based Decision Algorithm for Propositional Neighbourhood Logic. In *Proceedings of STACS 2007: 24th International Symposium on Theoretical Aspects of Computer Science*, vol. 4393 of LNCS, pp. 549-560, 2007.

[18] [BMS07b] Bresolin D. Montanari A and Sciavicco G. An Optimal Tableau-based Decision Procedure for Right Propositional Neighbourhood Logic. *Journal of Automated Reasoning*, vol 38(1-3), pp. 173-199, 2007.

[19] [BRC00] Barua R., Roy S. and Chaochen Z. Completeness of Neighbourhood Logic. *Journal of Logic and Computation*, vol. 10, num. 2, pp. 271-295, 2000.

[20] [Bur82] Burges J. P. Axioms for tense logic II: Time periods. *Notre Dame Journal of Formal Logic*, v. 23, pp. 375-383, 1982.

[21] [CE81] Clarke E. and Emerson E. A. Synthesis of synchronization skeletons for branching time temporal logic. In *Proceedings of IBM workshop on Logic of Programs*, pp. 52-71, Springer, 1981.

[22] [CG96] Campos S. and Grumberg O. Selective Quantitative Analysis and Interval Model Checking: Verifying Different Facets of a System. *Proceedings of the 8th International Conference on Computer Aided Verification, Lecture Notes in Computer Science*, Springer, num. 1102, pp. 257-268, 1996.

[23] [CGP00] Clarke E., Grumberg O. and Peled D. A. *Model Checking*. MIT Press, 2000.

[24] [CH98] Chaochen Z. and Hansen M. An Adequate First Order Interval Logic. *Compositionality: the Significant Difference*, *Lecture Notes in Computer Science, Springer*, num. 1536, pp. 584-608, 1998.

[25] [CH04] Chaochen Z. and Hansen M. Duration Calculus. A Formal Approach to Real-Time Systems. *EATCS Series of Monographs in Theoretical Computer Science*, Springer, 2004.

[26] [Cha99] Chaochen Z. Duration Calculus: A Logical Approach to Real-Time Systems. *Proceedings of the Annual Conference of the European Association for Computer Science Logic, Lecture Notes in Computer Science*, Springer, num. 1548, pp. 1-7, 1999.

[27] [Cho95] Chomicki J. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Transactions on Database Systems*, vol. 20, pp. 149-186, 1995.

[28] [CHR91] Chaochen Z., Hoare C. and Ravn A. P. A Calculus of Durations. *Information Processing Letters*, vol. 40, pp. 269-276, 1991.

[29] [CHR93] Chaochen Z., Hansen M. and Ravn A. An Extended Duration Calculus for Real-time Systems. *Hybrid Systems, Lecture Notes in Computer Science*, Springer, num. 736, pp. 36-59, 1993.

[30] [CHS93] Chaochen Z., Hansen M. and Sestoft P. Decidability and Undecidability results for Duration Calculus. In *Proceedings of the 10th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science,* Springer, num. 665, pp. 58-68, 1993.

[31] [CHX95] Chaochen Z., Hung D. V. and Xiaoshan L. *A* Duration Calculus with Infinite Intervals. *Fundamentals of Computation Theory, Lecture Notes in Computer Science*, Springer, num. 965, pp. 16-41, 1995.

[32] [CP93] Crouch R. and Pulman S. G. Time and Modality in a Natural Language Interface to a Planning System. *Artificial Intelligence*, vol. 63, num. 1-2, pp. 265-304, 1993.

[33] [CR03] Chagrov A.V. and Rybakov M.N. How Many Variable Does One Need to Prove PSPACE-hardness of Modal Logics. In *Advances in Modal Logic*, vol. 4, pp. 71-82, 2003.

[34] [CT98] Chomicki J. and Toman D. *T*emporal logics in information systems. In *Logics for Database and Information Systems*, Kluwer, 1998.

[35] [CX94] Chaochen Z. and Xiaoshan L. A Mean Value Calculus of Durations. *A Classical Mind: Essays in Honour of C.A.R. Hoare, Prentice Hall International Series in Computer Science*, pp. 431-451, 1994.

[36] [DG00] Demri S. and Gore R. An $O((n.logn)^3)$- Time Transformation from Grz into Decidable Fragments of Classical First-Order Logic. In *Automated Deduction in Classical and Non-Classical Logics*, vol.1761 of LNAI, pp. 153-167, 2000.

[37] [Dow79] Dowty D. *Word Meaning and Montague Grammar*. Dordrecht: D. Reidel, 1979.

[38] [Dut95] Dutertre B. On First-order Interval Temporal Logic. *Technical report, no. CSD-TR-94-3*, Department of Computer Science, Royal Holloway College, University of London, 1995.

[39] [EC80] Emerson E. A. and Clarke E. M. Characterizing correctness properties of parallel programs using fixpoints. In *Proceedings of 7th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science*, Springer, no. 85, pp. 169-181, 1980.

[40] [EC82] Emerson E. A. and Clarke E. M. *U*sing branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming* 2, pp. 241-266, 1982.

[41] [EH82] Emerson E. A. and Halpern, J. Y. Decision Procedures and expressiveness in the temporal logic of branching time. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pp. 169-180, 1982.

[42] [EH86] Emerson E. A. and Halpern, J. Y. 'Sometimes' and 'Not Never' Revisited: On Branching versus Linear Time Temporal Logic. *JACM*, vol. 33, no. 1, pp. 151-178,1986.

[43] [EJ88] Emerson E. A. and Jutla C.S. The complexity of free automata and logics of programs. In *Proceedings of the 29th Annual IEEE-CS Symposium on Foundations of Computer Science*, pp. 328-337, 1988.

[44] [Eme95] Emerson E. A. Temporal and Modal Logic. *Handbook of Theoretical Computer Science*, North-Holland Pub. Co., 1995.

[45] [FHMV96] Fagin R., Halpern J., Moses Y. and Vardi M. *Reasoning About Knowledge.* MIT Press, 1996.

[46] [Fis96a] Fisher M. An introduction to executable temporal logics. *Knowledge Engineering Review*, vol. 11, pp. 43-56, 1996.

[47] [Fis96b] Fisher M. A temporal semantics for concurrent MetateM. *Journal of Symbolic Computation*, vol. 22, pp. 627-648, 1996.

[48] [FM94] Fiaderio J.L. and Maibum T. Action Refinement in a Temporal Logic of Objects. *Temporal Logic, Lecture Notes in Computer Science*, Springer , vol. 827, 1994.

[49] [Fra96] Fraenzle M. Synthesizing Controllers from Duration Calculus. *Formal Techniques in Real-Time and Fault-Tolerant Systems, Lecture Notes in Computer Science*, vol. 1135, pp. 168-187, 1996.

[50] [Gal84] Galton A. *The Logic of Aspect*. Claredon Press, Oxford, 1984.

[51] [Gal90] Galton A. A Critical Examination of Allen's Theory of Action and Time. *Artificial Intelligence*, vol. 42, pp. 159-198, 1990.

[52] [GD99] Guelev D. and Dang V. H. On the Completeness and Decidability Duration Calculus with Iteration. *Advances in Computer Science, Lecture Notes in Computer Science*, vol. 1742, pp. 139-150, 1999.

[53] [GHR94] Gabbay D., Hodkinson I. and Reynolds M. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Clarendon Press, vol.1, Oxford, 1994.

[54] [GKWZ02] Gabbay D., Kurucz A., Wolter F. and Zakharyaschev M. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier, 2002.

[55] [GMS03a] Goranko V., Montanari A. and Sciavicco G. Propositional Interval Neighbourhood Temporal Logics. *Journal of Universal Computer Science*, vol. 9, num. 9, pp. 1137-1167, 2003.

[56] [GMS03b] Goranko V., Montanari A. and Sciavicco G. A General Tableau Method for Propositional Interval Temporal Logics. *Proceedings of the International Conference Tableaux* 2003, Lecture Notes in Artificial Intelligence, Springer, pp. 102-116, 2003.

[57] [GMS04] Goranko V., Montanari A. and Sciavicco G. A Road Map of Interval Temporal and Duration Calculi. *Journal of Applied Non-Classical Logics*, vol. 14, 2004.

[58] [Gol87] Goldblatt R. *Logics of time and computation*. Stanford Center for the study of Language and Information, 1987.

[59] [GPSS80] Gabbay D., Pnueli A., Shelah S. and Stavi J. On the temporal analysis of fairness. In *Proceedings 7th ACM Symposium Principles of Programming Languages*, pp. 163-173, 1980.

[60] [Gra99] Graedel E. Decision procedures for guarded logics. In *CADE'99, Lecture Notes in Artificial Intelligence*, Springer, vol. 1632, pp. 31-51, 1999.

[61] [GRF00] Gabbay D., Reynolds M. and Finger M. Temporal Logic: *Mathematical Foundations and Computational Aspects*. Clarendon Press, vol.2, Oxford, 2000.

[62] [Gue00a] Guelev D. A Complete Proof System for First Order Interval Temporal Logic with Projection. *no. 202*, UNU/IIST, 2000.

[63] [Gue00b] Guelev D. Probabilistic Neighbourhood Logic. *Proceedings of the 6th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems, Lecture Notes in Computer Science*, vol. 1926, pp. 264-275, 2000.

[64] [Ham71] Hamblin C. L. Instants and intervals. *Stadium Generale*, vol. 27, pp. 127-134, 1971.

[65] [Han94] Hansen M. Model-Checking Discrete Duration Calculus. *Formal Aspects of Computing*, vol. 6, pp. 826-845, 1994.

[66] [HC92] Hansen M. R. and Chaochen Z. Semantics and Completeness of Duration Calculus. *Real-Time: Theory in Practice, Lecture Notes in Computer Science,* Springer, num. 600, pp. 209-225, 1992.

[67] [HC97] Hansen M. and Chaochen Z. Duration Calculus: Logical Foundations. *Formal Aspects of Computing*, vol. 9, pp. 283-330, 1997.

[68] [HMM83] Halpern J. Y., Manna Z., and Moszkowski B. *A* high-level semantics based on interval logic. In *Proceedings of 10th ICALP*, pp. 278-291, 1983.

[69] [Hol91] Holzmann G.J. *Design and Validation of Computer Protocols.* Prentice Hall, New Jersey, 1991.

[70] [HPS83] Harel D., Pnueli A. and Stavi Y. Process logic: Expressiveness, decidability, completeness. *JCSS*, vol. 25, pp. 145-180, 1983.

[71] [HS91] Halpern J. Y. and Shoham Y. A Propositional Modal Logic of Time Intervals. *Journal of the ACM*, vol. 38, num. 4, pp. 935-962, 1991.

[72] [HS94] Hwang C. H. and Schubert L. K. Interpreting tense, aspect and time adverbials. *Proceedings of the First International Conference on Temporal Logic, Lecture Notes in Computer Science*, Springer, vol. 827, pp. 238-274, 1994.

[73] [Hum79] Humberstone L. Interval semantics for tense logic: Some remarks. *Journal of Philosophical Logic*, vol. 8, pp. 171-196, 1979.

[74] [HWZ00] Hodkinson I., Wolter F. and Zakharyaschev M. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, vol. 106, pp. 85-134, 2000.

[75] [Kam68] Kamp H. *Tense Logic and the Theory of Linear Order*. Ph.D. Thesis, University of California, LA, 1968.

[76] [Kam79] Kamp H. Events, Instants and temporal reference. *Semantics from Different Points of view*, Springer, pp. 376-417, 1979.

[77] [Kon06] Konur S. A Decidable Interval Temporal Logic for Events and States. *TIME 2006, International Symposium on Temporal Representation and Reasoning, IEEE Computer Society Press*, 2006.

[78] [KP86] Kornatzky Y. and Pinter S. An extension to partial order temporal logic (POTL). *EE Publication* no. 596, Department of Electrical Engineering, Technion-Israel Institute of Technology, 1986.

[79] [Kro87] Kroger F. *Temporal Logic of programs*. Springer, 1987.

[80] [Lad87] Ladkin P. Logical Time Pieces. *AI Expert*, pp. 58-67, 1987.

[81] [Lam80] Lamport L. Sometimes is Sometimes 'Not Never': On the temporal logic of programs. *7th Annual ACM Symposium on Principles of Programming Languages*, pp. 174-185, 1980.

[82] [Lod00] Lodaya K. Sharpening the Undecidability of Interval Temporal Logic. *Proceedings of 6th Asian Computing Science Conference, Lecture Notes in Computer Science*, Springer, num. 1961, pp. 290-298, 2000.

[83] [LP00] Lichtenstein O. and Pnueli A. Propositional Temporal Logics: Decidability and Completeness. *Logic Journal of the IGPL*, vol. 8, no.1, pp.55-85, Oxford University Press, 2000.

[84] [LPZ85] Lichtenstein O., Pnueli A. and Zuck L. The Glory of the Past. In *Proceeding Conferences on Logics of Programs, Lecture Notes in Computer Science*, Springer, vol. 193, pp. 196-218, 1985.

[85] [LS94] Laroussinie F. and Schnoebelen P. A hierarchy of temporal logics with past. In *Proceedings STACS 94, Lecture Notes in Computer Science*, Springer, no. 775, pp. 47-48, 1994.

[86] [LSWZ02] Lutz C., Sturm H., Wolter F. and Zakharyaschev M. A tableau decision algorithm for modalized ALC with constant domains. *Studia Logica*, vol. 72, pp. 199-232, 2002

[87] [McD82] McDermott D. A temporal Logic for reasoning about process and plans. *Cognitive Science*, vol. 6, pp. 101-155, 1982.

[88] [Mer92] Merz. Decidability and incompleteness results for first-order temporal logics of linear time. *Journal of Applied Non-classical Logic*, vol. 2, pp. 139-156, 1992.

[89] [Mos83] Moszkowski B. *Reasoning about Digital Circuits*. PhD Thesis, Stanford University, Computer Science Department, 1983.

[90] [Mos00a] Moszkowski B. An Automata-Theoretic Completeness Proof for Interval Temporal Logic. *Proceedings of the 27th International Conference on Automata, Languages and Programming, Lecture Notes in Computer Science*, Springer, no. 1853, pp. 223-234, 2000.

[91] [Mos00b] Moszkowski B. A complete axiomatization of Interval Temporal Logic with infinite time. *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press*, pp. 242-251, 2000.

[92] [MP81a] Manna Z. and Pnueli A. Verification of concurrent programs: The temporal framework. In *the correctness problem in computer science*, pp. 215-273, Academic Press, New York, 1981.

[93] [MP81b] Manna Z. and Pnueli A. Verification of concurrent programs: Temporal proof principles. *Lecture Notes in Computer Science*, Springer, vol. 131, pp. 200-252, 1981.

[94] [MR99] Marx M. and Reynolds M. Undecidability of Compass Logic. *Journal of Logic and Computation*, vol. 9, num. 6, pp. 897-914, 1999.

[95] [MSV02] Montanari A., Sciavicco G. and Vitacolonna N. Decidability of Interval Temporal Logics over Split-frames via Granularity. In *Proceedings of the 8th Europian Conference on Logics in AI*, vol. 2424 of LNAI, pp. 259-270, 2002.

[96] [OH95] Ohrstrom P. and Hasle P. F. Temporal Logic: From Ancient Ideas to Artificial Intellgience. *Kluwer Academic Publishers*, Dordrecht, The Netherlands, 1995.

[97] [Pan01] Pandya P.K. Specifying and Deciding Quantified Discrete-Time Duration Calculus Formulas using DCVALID. *Workshop on Real-Time Tools*, Aalborg, 2001.

[98] [Par78] Parikh R. A decidability result for second order process logic. In *Proceedings of 19th FOCS*, IEEE, pp. 177-183, 1978.

[99] [Par94] Parsons T. *Events in the Semantics of English: A Study of Subatomic Semantics*. MIT Press, 1994.

[100] [Pen95] Penczek W. Branching time and partial order in temporal logics. In *Time and Logic: a computational approach*, pp. 179-228, UCL Press, London, 1995.

[101] [PH05] Pratt-Hartman I. Temporal Prepositions and their logic. *Artificial Intelligence,* vol. 166(1-2), pp. 1-36, 2005.

[102] [PHB93] Pratt-Hartmann I. and Bree D. The Expressive Power of the English Temporal Preposition System. *Technical Report*, *UMCS-93-1-7*, University of Manchester.

[103] [Pli97] Pliuskevicius R. On the completeness and decidability of a restricted first-order linear temporal logic. In *Lecture Notes in Computer Science*, Springer, num. 1289, pp. 241-254, 1997.

[104] [Pnu85] Pnueli A. Linear and Branching Structures in the Semantics and Logics of Reactive Systems. *Proceedings of the 12th ICALP*, pp. 15-32, 1985.

[105] [PPH98] Penix J., Pecheur C. and Havelund K. Using Model Checking to Validate AI Planner Domain Models. *23rd Annual Software Engineering Workshop*, NASA Goddard, 1998.

[106] [Pra76] Pratt. V. R. Semantical considerations on Floyd-Hoare logic. In *Proceedings of 17th FOCS, IEEE*, pp. 109-121, 1976.

[107] [Pra79] Pratt V. R. Process Logic. In *Proceedings of 6th POPL, ACM,* pp. 93-100, 1979.

[108] [PW84] Pinter S. and Wolper P. A temporal logic for reasoning about partially ordered computations. In *Proceedings of the 3rd Symposium on Principles of Distributed Computing*, pp. 28-37, 1984.

[109] [Rab98] Rabinovich A. Non-Elementary Lower Bound for Propositional Duration Calculus. *Information Processing Letters*, vol. 66, pp. 7-11, 1998.

[110] [Rey96] Reynolds M. Axiomating first-order temporal logic: until and since over linear time. *Studia Logica*, vol. 57, pp. 279-302, 1996.

[111] [Rey01] Reynolds M. An axiomatization of full computation tree logic. *Journal of Symbolic Logic*, vol. 66, pp. 1011-1057, 2001.

[112] [Rey05] Reynolds M. An axiomatization of PCTL*. *Journal of Information and Computation*, Elsevier,vol. 201, pp. 72-119, 2005.

[113] [RG93] Rao A. and Georgeff M. A model-theoretic approach to the verification of situated reasoning systems. In *Proceedings of IJCAI*, 1993.

[114] [Rop80] Roper P. Intervals and tenses. *Journal of Philosophical Logic*, pp. 451-469, 1980.

[115] [SC85] Sistla A.P. and Clarke E.M. The complexity of propositional linear temporal logic. *Journal of ACM*, vol. 32, pp. 733-749, 1985.

[116] [Sim87] Simons P. *Parts, A Study in Ontology.* Claredon Press, Oxford, 1987.

[117] [Spi00] Spielmann M. Verification of relational transducers for electronic commerce. In *PODS'2000, ACM*, pp. 92-103, 2000.

[118] [SVW87] Sistla A. P., Vardi Y. and Wolper P. The complementation problem for Buchi automata with applications to temporal logics. *Theoretical Computer Science*, vol. 49, pp. 217-237, 1987.

[119] [Sza95] Szalas A. Temporal logic of Programs: standard approach. In *Time and logic: A Computational Approach*, pp. 1 -50, UCL Press, London, 1995.

[120] [Ter96] Ter Meulen A. G. *Representing Time in Natural Language*. MIT Press, Cambridge, 1996.

[121] [vB83] van Benthem J. F. *The Logic of Time*. Dordrecht, 1983.

[122] [Ven67] Vendler Z. *Linguistics and Philosophy*. Cornell University Press, 1967.

[123] [Ven90] Venema Y. Expressiveness and Completeness of an Interval Tense Logic. *Notre Dame Journal of Formal Logic*, vol. 31, num. 4, pp. 529-547, 1990.

[124] [Ven91] Venema Y. A Modal Logic for Choppping Intervals. *Journal of Logic and Computation*, vol. 1, pp. 453-476, 1991.

[125] [Ven98] Venema Y. *Temporal Logic*. Blackwell Guide to Philosophical Logic, Blacwell Publishers, 1998.

[126] [Wal47] Walker A. G. *Durees et instants*. La Revue Scientifique, num. 3266, 1947.

[127] [Whi72] Whitrow G. J. Reflections on the Concept of Time. *The Study of Time*, Springer, vol. 1, pp. 1-11, 1972.

[128] [WZ99] Wolter F. and Zakharyaschev M. Modal description logics: modalizing roles. *Fundamenta Informaticae*, vol. 39, pp. 411-438, 1999.

[129] [WZ00] Wolter F. and Zakharyaschev M. Temporalizing description logics. In *Frontiers of Combining Systems II*, pp. 379-401, Research Studies Press LTD, UK, 2000.

[130] [WZ02] Wolter F. and Zakharyaschev M. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Apllied Logic*, vol. 118, pp. 133-145, 2002.