

# Cutting Diamonds: A Temporal Logic with Probabilistic Distributions

Alisa Kovtunova, Rafael Peñaloza

KRDB Research Centre, Free University of Bozen-Bolzano, Italy  
{alisa.kovtunova,rafael.penalaza}@unibz.it

## Abstract

Temporal logics, as formalisms capable of expressing properties evolving over time, have been successfully employed to represent and verify processes in many different settings. A common staple in most of these logics is the eventuality (or diamond) constructor, which expresses that some property will hold at some point in the future. While useful in practice, the specification of the diamond operator is too rough; indeed, one often has an idea—albeit uncertain—about *when* the property may hold. We introduce TLD, an extension of linear temporal logic (LTL) that refines the diamond operator with a new constructor expressing a probability distribution for the time until the property is observed. We study the main properties of this logic and describe methods for deciding satisfiability, and performing probabilistic inferences in a restricted version of the logic. These methods rely on known properties of LTL.

## Introduction

Temporal logics have been widely studied as formalisms for expressing and dealing with properties that evolve over time. Over the years, variants of these formalisms have been successfully employed to represent and verify processes from a large variety of domains. Perhaps one of the main success stories from the field comes from the area of model checking, where the goal is to verify that a system specification satisfies a given temporal property (Clarke, Emerson, and Sistla 1986).

In the literature, it is possible to find many different definitions of temporal logics (Pnueli 1986), depending on the view one has on time—discrete or continuous (Koymans 1990; Maler, Nickovic, and Pnueli 2005), linear or branching (Emerson and Srinivasan 1989), with or without past (Markey 2003; 2004), etc.—the expressivity (or constructors) allowed—eventually, always, next, etc.—and even the kind of underlying logical formalism that is being extended with time—e.g., propositional logic, description logics (Lutz, Wolter, and Zakharyashev 2008), or even databases (Chomicki 1994). Among all these logics, linear temporal logic (LTL) (Pnueli 1977) remains a popular choice for dealing with discrete linear time properties and specifications.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

An important staple in LTL, which is also commonly observed in many other related logics, is the *eventuality* constructor (also known as *diamond*). In a nutshell, the diamond expresses that a given property will be *eventually* observed. This simple constructor is very useful in the specification of desirable properties, but is often too rough in relation to our knowledge of the world: it does not allow for a finer description of the potential waiting time until the eventuality is fulfilled, unless one can provide some certainty guarantees for this time. For example, when stopping at a red traffic light, we know that it will *eventually* turn green; but we are also almost certain that this event will happen within the next five minutes; still, under some circumstances, we might have to wait six, or more minutes. Similarly, when we buy a new washing machine, we know that it will *eventually* break, but we do not expect that to happen either in the very near (e.g., tomorrow), nor in the very far future (in 500 years). Indeed, in this case we would expect the (perhaps subjective) probability of the machine breaking to increase with time until it reaches a maximum point from which it starts to decrease. To handle this information, it is necessary to allow the expressivity of probability distributions over the time that one must wait for the property to hold.

In this paper we introduce TLD,<sup>1</sup> a new probabilistic extension of LTL that refines the diamond constructor with the new, freshly cut, operator  $\blacklozenge$ . This new operator is parameterized with a discrete probability distribution that can be used to specify the likelihood of observing the property of interest, for the first time, at each possible point in time. Hence, for the traffic light example, one can specify a distribution that gives a very high probability only to the time covering the first five minutes, with this probability decreasing very fast as time distances from the five-minute mark in the future. In the case of the washing machine, one can specify an almost bell-shaped infinite distribution such as the Poisson, which assigns a positive—albeit, rapidly decreasing after its mean or expected value—probability of occurrence to every point in time.

We provide a formal definition of our new logic TLD and its semantics which is based on the idea of *subjective* probabilities; that is, uncertainty refers to a view in which many different scenarios (or *worlds*) are possible, and the proba-

<sup>1</sup>TLD stands for *temporal logic with probabilistic distributions*.

bilities reflect the likelihood of each of these worlds of being observed. In our case, a world would be a possible evolution of a system; that is, an LTL interpretation. We show that the  $\heartsuit$  constructor is very expressive in the sense that it can describe properties that can only be satisfied on uncountable interpretations. Hence, we introduce a restricted version, called  $\text{TLD}^-$ , that has the *countable-model property*; i.e., every satisfiable  $\text{TLD}^-$  formula has a countable model. Within this sub-logic, we present effective methods for deciding satisfiability, and computing probabilities of different events. Moreover, we show how to compute the *expected* (essentially, the average) time required until a desired property is observed.

## Preliminaries

We briefly introduce the basics of probability theory and linear temporal logic (LTL).

### Probability Theory

We start by providing the basic notions of probability needed for this paper. For a deeper study on probabilities, we refer the interested reader to (Billingsley 1995).

Let  $\Omega$  be a set called the *sample space*. A  $\sigma$ -algebra over  $\Omega$  is a class  $\mathcal{F}$  of subsets of  $\Omega$  that contains the empty set, and is closed under complements and under countable unions. A *probability measure* is a function  $\mu : \mathcal{F} \rightarrow [0, 1]$  such that  $\mu(\Omega) = 1$ , and for any countable collection of pairwise disjoint sets  $E_i \in \mathcal{F}$ ,  $i \geq 1$ , it holds that

$$\mu\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mu(E_i).$$

The probability of a set  $E \in \mathcal{F}$  is  $\int_{\omega \in E} \mu d\omega$ , where the integration is made w.r.t. the measure  $\mu$ . For special cases where this integration is easily defined, see below.

A usual case is when  $\Omega$  is the set  $\mathbb{R}$  or all real numbers, and  $\mathcal{F}$  is the standard Borel  $\sigma$ -algebra over  $\mathbb{R}$ ; that is, the smallest  $\sigma$ -algebra containing all open intervals in  $\mathbb{R}$ . In this case,  $\mu$  is called a *continuous* probability measure, and the integration defining the probability of a set  $E$  corresponds to the standard Riemann integration.

If  $\Omega$  is a countable (or finite) set, the standard  $\sigma$ -algebra is formed by the power set of  $\Omega$ , and for any probability measure  $\mu$  over  $\Omega$  and set  $E \subseteq \Omega$ ,  $P(E) = \sum_{\omega \in E} \mu(\omega)$ ; that is, the probability of a set is the sum of the probabilities of the elements it contains. In this case,  $\mu$  is called a *discrete* probability measure. If, in addition,  $\mu(\omega) > 0$  for all  $\omega \in \Omega$ , then  $\mu$  is *complete*. When  $\Omega$  is the set of all natural numbers  $\mathbb{N}$ , we specify the distribution  $\mu$  as a function  $\mu : \mathbb{N} \rightarrow [0, 1]$ . In this case, the *expectation* of  $\mu$  is  $E[\mu] := \sum_{n \in \mathbb{N}} n \cdot \mu(n)$ . Intuitively, the expectation estimates the value that one should predict to see under  $\mu$ .

Two simple examples of complete discrete distributions are the *geometric* and the *negative binomial* (also known as *Pascal*) distributions. The geometric distribution, defined by a parameter (the probability of success)  $p \in (0, 1)$ , is defined, for every  $i \in \mathbb{N}$ , by  $\mu(i) = (1 - p)^{i-1} p$ . This distribution describes the probability of observing the first success in a repeated trial of an experiment at time  $i$ . The negative

binomial generalizes the geometric by counting the time until a pre-specified number  $n$  of successes has been observed.

### LTL

LTL is a temporal logic that extends propositional logic with the discrete temporal operators  $\bigcirc$  (next) and  $\mathcal{U}$  (until). More precisely, given a set  $\mathcal{V}$  of *propositional variables*, *LTL formulas* are defined by the following grammar, where  $x \in \mathcal{V}$ :

$$\varphi ::= x \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi\mathcal{U}\varphi.$$

An *LTL interpretation* is a function  $I : \mathbb{N} \rightarrow 2^{\mathcal{V}}$  that maps every  $n \in \mathbb{N}$  to a set  $I(n) \subseteq \mathcal{V}$ . Given an interpretation  $I$ , a formula  $\varphi$ , and  $n \in \mathbb{N}$ , the satisfaction relation  $I, n \models \varphi$  is defined inductively as follows:  $I, n \models x$  iff  $x \in I(n)$ ;  $I, n \models \neg\varphi$  iff  $I, n \not\models \varphi$ ;  $I, n \models \varphi_1 \wedge \varphi_2$  iff  $I, n \models \varphi_1$  and  $I, n \models \varphi_2$ ;  $I, n \models \bigcirc\varphi$  iff  $I, n+1 \models \varphi$ ; and  $I, n \models \varphi_1\mathcal{U}\varphi_2$  iff  $\exists m \geq n. I, m \models \varphi_2$  and for all  $i, n \leq i < m$ ,  $I, i \models \varphi_1$ . The interpretation  $I$  is a *model* of  $\varphi$  ( $I \models \varphi$ ) iff  $I, 0 \models \varphi$ .

We denote as  $\top$  any propositional tautology, and use the common abbreviations  $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$ ,  $\heartsuit\varphi := \top\mathcal{U}\varphi$ , and  $\square\varphi := \neg\heartsuit\neg\varphi$ . Of particular interest for this work is the operator  $\heartsuit$ , commonly read as *eventually*. In a nutshell, the formula  $\heartsuit\varphi$  is true at some point in time if there is some future time where  $\varphi$  is satisfied. However, there is no notion of how soon or how late that time will take place; that is, the satisfaction of the  $\heartsuit$  may be delayed indefinitely.

It is well known that satisfiability of an LTL formula  $\varphi$  can be decided by checking the emptiness of a (generalized) Büchi automaton (GBA) that accepts all models of  $\varphi$ . Briefly, given such a formula  $\varphi$ , let  $\text{sub}(\varphi)$  be the set of all subformulas of  $\varphi$ , and  $\text{cl}(\varphi)$  be the closure under negation of the set  $\text{sub}(\varphi) \cup \{\bigcirc(\psi_1\mathcal{U}\psi_2) \mid \psi_1\mathcal{U}\psi_2 \in \text{sub}(\varphi)\}$ . If  $\text{At}$  is the set of all the maximally consistent subsets of  $\text{cl}(\varphi)$ , then the GBA  $\mathcal{A}_\varphi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$ , where  $Q = \text{At}$ ,  $\Sigma = 2^{\mathcal{V}}$ ,  $\text{Ini} = \{s \in \text{At} \mid \varphi \in s\}$ , and

$$\begin{aligned} \Delta &= \{(s, V, s') \mid s \cap \mathcal{V} = V, \bigcirc\psi \in s \rightarrow \psi \in s'\} \\ \mathcal{F} &= \{\{s \mid \psi' \in s \vee \psi\mathcal{U}\psi' \notin s\} \mid \psi\mathcal{U}\psi' \in \text{sub}(\varphi)\} \end{aligned}$$

accepts all models of  $\varphi$  (Vardi and Wolper 1994). Intuitively, the states in a successful run of  $\mathcal{A}_\varphi$  correspond to the subformulas of  $\varphi$  that are satisfied by the accepted model in every point in time.

## Distributions over Time

When dealing with temporal information, and in particular with eventualities, one may be interested in understanding *when* will such an eventuality be resolved. Usually, one would have some knowledge about the likelihood of observing the event at different points in time, which may be expressed in the form of a discrete probability distribution  $\delta$ . For example, if a quality control process is triggered when a pre-specified number (say, 5) of defective items have been produced, and each item may be independently defective with some probability (e.g., 0.5), the waiting time until the quality control process is triggered can be modelled through a negative binomial distribution with parameters 5 and 0.5. In that case, it is more likely that the statement “eventually

the quality control process is triggered” is satisfied by a witness at time 5 than, say, at time 10. Thus, we want to extend the notion of  $\diamond$  to handle this and potentially other kinds of probabilistic information derived from the distribution.

The *temporal logic with probabilistic distributions* (TLD) extends LTL with a new constructor  $\diamond_\delta\varphi$ , called the *distribution eventuality*, that expresses that the time until the formula  $\varphi$  is first observed has distribution  $\delta$ . Formally, TLD formulas are built from LTL formulas  $\varphi$  by the grammar rule

$$\Phi ::= \varphi \mid \diamond_\delta\varphi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \bigcirc\Phi \mid \mathcal{U}\Phi \mid \square\Phi,$$

where  $\delta$  is a probability distribution over  $\mathbb{N}$ . When the specific distribution  $\delta$  is not relevant, we will often disregard the subindex and write e.g.,  $\diamond\psi$ . Notice that we disallow  $\diamond$  to appear in the scope of any explicit or implicit negation, since  $\neg\diamond\varphi$  simply expresses that the time until  $\varphi$  is observed has a distribution different from  $\delta$ , which is an uninformative statement. However, we explicitly include the abbreviations ( $\vee, \square$ ) that are lost by the lack of negation. The semantics of this logic is based on the *multiple-world* approach.

**Definition 1.** A TLD interpretation is a pair  $\mathcal{P} = (\mathcal{I}, \mu)$ , where  $\mathcal{I}$  is a set of LTL interpretations and  $\mu$  is a probability distribution over  $\mathcal{I}$ . Given a set of LTL interpretations  $\mathcal{I}$ , an LTL formula  $\varphi$ , and  $n \in \mathbb{N}$ , let  $\mathcal{I}_n^\varphi := \{I \in \mathcal{I} \mid I, n \models \varphi\}$ . The satisfaction relation  $\mathcal{P}, n \models \Phi$  is defined by induction as follows. For every  $I \in \mathcal{I}$ ,

- $I, n \models \diamond_\delta\varphi$  iff for all  $i \geq 0$ ,  $\mu(\mathcal{I}_{n+i}^\varphi \setminus \bigcup_{j=0}^{i-1} \mathcal{I}_{n+j}^\varphi) = \delta(i)$ ,
- $\mathcal{P}, n \models \Psi$  iff for every  $I \in \mathcal{I}$  it holds that  $I, n \models \Psi$ ,

where the satisfaction relation on the LTL constructors ( $\wedge, \bigcirc, \mathcal{U}, \square$ ) is defined in the usual manner as for LTL. If  $\mathcal{P}, 0 \models \Phi$ , we call  $\mathcal{P}$  a *model* of  $\Phi$ .  $\Phi$  is *satisfiable* if it has a model. The interpretation  $\mathcal{P} = (\mathcal{I}, \mu)$  is *countable* if the set  $\mathcal{I}$  contains countably many LTL interpretations.

The following simple example showcases the semantics of the probabilistic constructor  $\diamond$ .

*Example 2.* Let  $\delta$  be the geometric distribution with parameter  $\frac{1}{2}$ ; that is,  $\delta(i) = 1/2^{i+1}$  for all  $i \in \mathbb{N}$ , and  $a$  a propositional variable. For every  $n \in \mathbb{N}$ , let  $\mu(I_n) = \delta(n)$ , and  $I_n$  be the LTL interpretation such that

$$I_n(k) = \begin{cases} \emptyset & k < n \\ \{a\} & k \geq n; \end{cases}$$

that is,  $I_n$  makes the proposition  $a$  false in the first  $n - 1$  points in time, and true afterwards. Then the TLD interpretation  $(\{I_n \mid n \in \mathbb{N}\}, \mu)$  is a model for  $\diamond_\delta a$ .

In general, we cannot make any assumptions about the class  $\mathcal{I}$  of LTL interpretations that will be used when building a model for a TLD formula  $\Phi$ . In fact, there are simple formulas that only have uncountable models.

*Example 3.* Consider the TLD formula  $\Phi_0 := \square\diamond_\delta a$ , where  $\delta$  is the geometric distribution with parameter  $\frac{1}{2}$ . Since  $\Phi_0$  contains only one propositional variable  $a$ , every real number  $x \in [0, 1]$  can be seen as an LTL interpretation  $I_x$  where  $I_x(n) = \{a\}$  if the  $n + 1$ -st digit of the (standard) binary representation of  $x$  is 1, and  $I_x(n) = \emptyset$  otherwise.

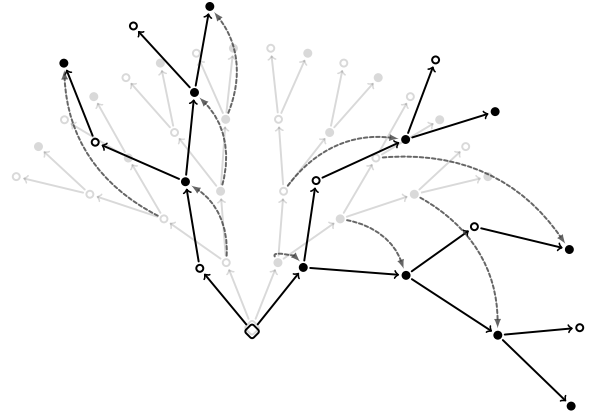


Figure 1: Embedding of the full binary tree (in gray) into the tree  $T$  (in black). Filled nodes are labelled with 0 and unfilled nodes with a 1.

Consider now  $\mathcal{P} = (\{I_x \mid x \in [0, 1]\}, \mu)$ , where  $\mu$  is the standard Borel measure that maps every interval in  $[0, 1]$  to its length. Then  $\mathcal{P} \models \Phi_0$ . To see this, it suffices to notice that for every  $n, i$ ,  $\mathcal{I}_{n+i}^a \setminus \bigcup_{j=0}^{i-1} \mathcal{I}_{n+j}^a$  is isomorphic to the set of all numbers whose binary representation starts with a chain of  $i$  zeroes, followed by a one, which is the interval  $[\frac{1}{2^{i+1}}, \frac{1}{2^i})$  of length  $\frac{1}{2^{i+1}} = \delta(i)$ .

Notice that the model  $\mathcal{P}$  provided in this example is not unique; in fact,  $\Phi_0$  has infinitely many models which can be obtained with the help of mixed measures giving a positive probability to interpretations  $I_x$  for some rational numbers  $x$ . However, all these models are uncountable.

**Theorem 4.** Let  $\mathcal{P} = (\mathcal{I}, \mu)$  be a TLD interpretation, and  $\delta(i) = 1/2^{i+1}$  for all  $i \in \mathbb{N}$ . If  $\mathcal{P} \models \square\diamond_\delta a$ , then  $\mathcal{I}$  is uncountable.

*Proof (Sketch).* Since the only propositional variable in the formula is  $a$ , every interpretation  $I \in \mathcal{I}$  is isomorphic to an infinite word over  $\{0, 1\}$  (that is, the binary representation of a real number in the unit interval). We can then represent  $\mathcal{I}$  as a binary tree  $T$ , where every node is labeled with a 0 or a 1 according to the binary encoding described before. By construction, every node labeled with a 0 must have two successors, labeled with each of the two values. Remove from  $T$  every node that cannot reach a node labeled with a 0. Then every node in  $T$  has a branching successor. This means that the full binary tree can be embedded in  $T$  (see Figure 1). Thus,  $T$  has at least as many paths as the full binary tree, which are uncountable. Since every path in  $T$  corresponds to an interpretation in  $\mathcal{I}$ , the result follows.  $\square$

Uncountable models are problematic from a computational point of view because they are harder to represent and handle finitely. Indeed, notice that the characterization of the models from Theorem 4 is far from trivial. Moreover, probability distributions over uncountable universes require more advanced mathematical tools. To avoid this issue for the moment, we consider a syntactic restriction of TLD that allows formulas of the form  $\diamond_\delta\varphi$  to be satisfied in at most one point

in time. Formally,  $\text{TLD}^-$  formulas are constructed through the grammar

$$\Phi ::= \varphi \mid \diamond_{\delta} \varphi \mid \Phi \wedge \varphi \mid \Phi \vee \varphi \mid \bigcirc \Phi \mid \varphi \mathcal{U} \Phi.$$

$\text{TLD}^-$  formulas contain at most one occurrence of the constructor  $\diamond$ , and this occurrence is always within the scope of only Boolean or temporal operators that need only one time point to be satisfied. This allows us to show that  $\text{TLD}^-$  has the countable-model property.

**Theorem 5.** *The  $\text{TLD}^-$  formula  $\Phi$  is satisfiable iff it has a countable model.*

*Proof.* Let  $\mathcal{P} = (\mathcal{I}, \mu)$  be a model of  $\Phi$ , and  $\diamond_{\delta} \varphi$  be the (only)  $\diamond$ -formula appearing in  $\Phi$  (if there is none, the result follows trivially). If there is no  $n \in \mathbb{N}$  such that  $\mathcal{P}, n \models \diamond_{\delta} \varphi$ , then for every  $I \in \mathcal{I}$ , the interpretation  $\mathcal{P}_I = (\{I\}, \mu_I)$ , where  $\mu_I(\{I\}) = 1$  is a model of  $\Phi$ .

Otherwise, if there is an  $n \in \mathbb{N}$  with  $\mathcal{P}, n \models \diamond_{\delta} \varphi$ , then by the semantics of  $\diamond$ , for every  $i \in \mathbb{N}$  such that  $\delta(i) > 0$  there exists an interpretation  $I_i \in \mathcal{I}$  with  $I_i, n+i \models \varphi$  and  $I_i, n+j \not\models \varphi$  for all  $0 \leq j < i$ . If there is more than one, just choose one as a representative. Then, the  $\text{TLD}$  interpretation  $\mathcal{P}' = (\{I_i \mid i \in \mathbb{N}\}, \mu')$ , where  $\mu'(\{I_i\}) = \delta(i)$  is a countable model of  $\Phi$ .  $\square$

## Deciding Satisfiability

For this section, we focus on the problem of deciding whether a given  $\text{TLD}^-$  formula  $\Phi$  is satisfiable. Recall that the constructor  $\diamond_{\delta}$  can appear at most once in a  $\text{TLD}^-$  formula. Hence, it suffices to focus on one discrete probability distribution  $\delta$ . For the rest of this section, we consider  $\delta$  to be an arbitrary but fixed complete distribution. To improve readability we will omit the subscript  $\delta$  from these formulas.

To start our analysis of satisfiability of  $\text{TLD}^-$  formulas, consider the simple case where  $\Phi = \diamond \varphi$  for some LTL formula  $\varphi$ . Already this case requires machinery beyond standard LTL. For example, the formula  $\Psi = \diamond(\diamond a)$  is unsatisfiable. To see this recall that any  $\text{TLD}$  model of  $\Psi$  must contain an LTL interpretation  $I$  such that  $I, 0 \models \neg(\diamond a)$  and  $I, 1 \models \diamond a$ . The former means that  $I, n \models \neg a$  for all  $n \geq 0$ , but the latter says that there must exist some  $m \geq 1$  such that  $I, m \models a$ , which is a contradiction.

In fact, for any model  $\mathcal{P} = (\mathcal{I}, \mu)$  of the formula  $\diamond \varphi$  and any  $n \in \mathbb{N}$ , there must exist an  $I_n \in \mathcal{I}$  such that  $I_n, n \models \varphi$  and  $I_n, k \not\models \varphi$  for all  $k < n$ . Another way to visualize this is to think that we allow points of time in the past, and for every  $n$  we should be able to find an interpretation  $I$  over the integers such that  $I, 0 \models \varphi$  and  $I, k \not\models \varphi$  for all  $-n < k < 0$ . This motivates the following theorem.<sup>2</sup>

**Theorem 6.** *The  $\text{TLD}$  formula  $\diamond \varphi$  is satisfiable iff the  $\text{LTL}_p$  formula  $\varphi \wedge \square_p \neg \varphi$  is satisfiable.*

*Proof.* If  $I$  is a model of  $\varphi \wedge \square_p \neg \varphi$ , then  $I, 0 \models \varphi$  and for all  $k < 0$ ,  $I, k \not\models \varphi$ . For each  $n \in \mathbb{N}$  we construct the LTL

<sup>2</sup>LTL interpretations are extended to integers in the obvious way:  $I$  maps every integer to a set of propositional variables. Such an interpretation  $I$  satisfies the formula  $\square_p \varphi$  if for every  $k < 0$ ,  $I, k \models \varphi$ .  $\text{LTL}_p$  is the extension of LTL with this past operator.

interpretation  $I_n : \mathbb{N} \rightarrow 2^V$  defined by  $I_n(k) := I(k-n)$  for all  $k \in \mathbb{N}$ . The  $\text{TLD}$  interpretation  $(\{I_n \mid n \in \mathbb{N}\}, \mu)$  where  $\mu(\{I_n\}) = \delta(n)$  for all  $n \in \mathbb{N}$  is a model of  $\diamond \varphi$ .

Conversely, it is known that every  $\text{LTL}_p$  formula  $\psi$  has a model iff it has an *ultimately periodic* model to the past; i.e., a model  $I$  such that, for some constants  $-2^{|\psi|} \leq n, k < 0$  it holds that for all  $m < n$ ,  $I(m) = I(m+k)$ . If  $(\mathcal{I}, \mu) \models \diamond \varphi$ , then for  $m > 4^{|\varphi|}$  there exists an interpretation  $I_m \in \mathcal{I}$  such that  $I_m, m \models \varphi$  and  $I_m, k \not\models \varphi$  for all  $0 \leq k < m$ . This interpretation can be extended to all negative numbers to satisfy  $\varphi \wedge \square_p \neg \varphi$ .  $\square$

Since deciding satisfiability of LTL formulas with past operators is the same as for standard LTL formulas (Kesten et al. 1993; Markey 2004), we obtain a tight complexity bound.

**Corollary 7.** *If  $\varphi$  is an LTL formula, deciding satisfiability of  $\diamond \varphi$  is PSpace-complete.*

We turn now our attention to the general case, where  $\Phi$  is an arbitrary  $\text{TLD}^-$  formula. Without loss of generality, assume that the only appearance of the constructor  $\diamond$  is of the form  $\diamond a$ , where  $a$  is a propositional variable. For example, the formula  $\diamond \varphi$  can be transformed into the equivalent  $\diamond a \wedge \square((a \wedge \varphi) \vee (\neg a \wedge \neg \varphi))$ . Given such a formula  $\Phi$ , let  $\Phi^{\downarrow}$  be the LTL formula obtained by substituting the (unique) occurrence of  $\diamond a$  with the tautology  $a \vee \neg a$ , and  $\Phi^-$  the LTL formula obtained by substituting  $\diamond a$  with the contradiction  $a \wedge \neg a$ .

Suppose first that  $\Phi^-$  is satisfiable; that is, it has an LTL model  $I \models \Phi^-$ . Then, clearly, the original formula  $\Phi$  is also satisfiable. Indeed, the  $\text{TLD}$  interpretation  $\mathcal{P} = (\{I\}, \mu)$  where  $\mu(\{I\}) = 1$  is a model of this formula, where  $\diamond a$  is never satisfied. As a simple example, any model of the LTL formula  $\varphi$  is also a model of  $\diamond a \vee \varphi$ . Since this case can be verified easily, we assume for the rest of this section that  $\Phi^-$  is not satisfiable. This means that every model must satisfy  $\diamond a$  at some point in time. The following lemma formalizes this.

**Lemma 8.** *Let  $\Phi$  be a  $\text{TLD}^-$  formula. Then, (i) if  $\Phi^-$  is satisfiable, then  $\Phi$  is satisfiable; and (ii) if  $\Phi^-$  is not satisfiable, then for every model  $\mathcal{P}$  of  $\Phi$  there exists an  $n \in \mathbb{N}$  such that  $\mathcal{P}, n \models \diamond a$ .*

To study satisfiability of  $\text{TLD}^-$  formulas, we first consider the special case where the probabilistic formula  $\diamond a$  is satisfied at the first point in time. We will extend the results to the general case later in this section.

**Definition 9.** A  $\text{TLD}^-$  formula  $\Phi$  is *originally satisfiable* if there is a model  $\mathcal{P}$  of  $\Phi$  such that  $\mathcal{P}, 0 \models \diamond a$ .

Obviously, if  $\Phi$  is originally satisfiable, then it is also satisfiable, but the converse does not hold in general. For instance, the formula  $\bigcirc \diamond a$  is satisfiable, but not originally satisfiable. In general, the following equivalence holds.

**Theorem 10.**  *$\Phi$  is originally satisfiable iff for every  $n \in \mathbb{N}$  there exists an LTL interpretation  $I_n$  such that (i)  $I_n \models \Phi^{\downarrow}$ , (ii)  $I_n, n \models a$ , and (iii) for all  $k, 0 \leq k < n, I_n, k \not\models a$ .*

*Proof (Sketch).* The result follows from the observation that if  $\mathcal{P} = (\mathcal{I}, \mu)$  is a  $\text{TLD}$  model of  $\Phi$ , then every  $I \in \mathcal{I}$  is an

LTL model of  $\Phi^\downarrow$ , which can be easily shown by structural induction on the shape of  $\Phi$ . The conditions (ii) and (iii) follow then from the semantics of the  $\blacklozenge$  operator.  $\square$

The formulation of this theorem shows a similarity to the case studied at the beginning of this section. However, contrary to the case of Theorem 6, we cannot simply use a past temporal operator to ensure original satisfiability. The reason for this is that the interpretations  $I_n$  may need to differ from each other.

*Example 11.* Consider the formula  $\Phi_1 = \blacklozenge a \wedge b \wedge \bigcirc(\neg b \mathcal{U} a)$ , which is originally satisfiable. In this case, the models of  $\Phi_1^\downarrow = b \wedge \bigcirc(\neg b \mathcal{U} a)$  must satisfy  $b$  at the first point in time, but then must satisfy  $\neg b$  (that is, violate  $b$ ) until an  $a$  is observed. In other words, for every model  $\mathcal{P} = (\mathcal{I}, \mu)$  of  $\Phi_1$  and every  $n > 0$  there must exist an interpretation  $I_n \in \mathcal{I}$  such that  $I_n, 0 \models \neg a \wedge b$ ,  $I_n, n \models a$ , and for all  $j, 0 < j < n$ ,  $I_n, j \models \neg a \wedge \neg b$ . This cannot be verified through a single satisfiability test on an LTL formula with past operator, since the chain of  $\neg b$  grows arbitrarily.

Recall that given an LTL formula  $\varphi$ , we can construct a GBA  $\mathcal{A}_\varphi$  that accepts all the LTL models of  $\varphi$ . Moreover, the states of this automaton represent the set of subformulas of  $\varphi$  that are satisfied at each point in time by this model. For a  $\text{TLD}^-$  formula  $\Phi$ , let  $\mathcal{A}_\Phi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$  be the generalized Büchi automaton that accepts all models of  $\Phi^\downarrow$ . Assume w.l.o.g. that  $\mathcal{A}_\Phi$  is *reduced* in the sense that all the states in  $Q$  appear in at least one successful run of  $\mathcal{A}_\Phi$ ; more precisely, that every state in  $Q$  is reachable from  $I$ , and from it one can reach a cycle that contains at least one state from each of the sets in  $\mathcal{F}$ . Any state that does not satisfy these requirements can be removed without modifying the language accepted by the automaton.

**Definition 12.** Given the  $\text{TLD}^-$  formula  $\Phi$ , and the reduced GBA  $\mathcal{A}_\Phi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$  that accepts all models of  $\Phi^\downarrow$ , the *origin automaton* of  $\Phi$  is the non-deterministic finite automaton (NFA)  $\mathcal{B}_\Phi = (Q, \{1\}, \Delta', \text{Ini}, F)$ , where

$$F = \{s \in Q \mid a \in s\},$$

$$\Delta' = \{(s, 1, s') \mid (s, V, s') \in \Delta, s \notin F\}.$$

Notice that  $\mathcal{A}_\Phi$  and  $\mathcal{B}_\Phi$  differ on the set of accepting states, but also on the class of languages they accept.  $\mathcal{A}_\Phi$  accepts infinite words, and its acceptance condition requires some types of states to be visited infinitely often. On the other hand  $\mathcal{B}_\Phi$  accepts only finite words over a unary alphabet, and the acceptance condition requires that the state reached after reading this word is in  $F$ , which is defined as the set of states containing  $a$ . Specifically, successful runs over  $\mathcal{B}_\Phi$  represent partial LTL models ending in a type containing the propositional variable  $a$ . In addition, types containing  $a$  do not have any outgoing transition in  $\mathcal{B}_\Phi$ . This means that any successful run  $\rho$  of the origin automaton of length  $n$  represents a prefix of an LTL model  $I_n$  of  $\Phi^\downarrow$  such that  $I_n, n \models a$  and  $I_n, k \not\models a$  for all  $0 \leq k < n$ . Thus, we get the following result.

**Theorem 13.** *The  $\text{TLD}^-$  formula  $\Phi$  is originally satisfiable iff the automaton  $\mathcal{B}_\Phi$  accepts the language  $1^*$ .*

*Proof.* If  $\Phi$  is originally satisfiable, then there is a model  $\mathcal{P} = (\mathcal{I}, \mu)$  of  $\Phi$  such that  $\mathcal{P}, 0 \models \blacklozenge a$ . Since  $\delta$  is complete, for every  $n \in \mathbb{N}$  there must exist some  $I_n \in \mathcal{I}$  such that  $I_n, n \models a$  and  $I_n, j \not\models a$  for all  $0 \leq j < n$ . Then, the first  $n$  transitions in the run accepting  $I_n$  form a successful run for  $1^n$  in  $\mathcal{B}_\Phi$ . Hence the automaton accepts  $1^*$ .

Conversely, every successful run of  $\mathcal{B}_\Phi$  of length  $n$ , defines a sequence of transitions in  $\mathcal{A}_\Phi$  that uses the same state. Since  $\mathcal{A}_\Phi$  is reduced, from the final state reached by  $\mathcal{B}_\Phi$ , it is possible to construct a successful run in  $\mathcal{A}_\Phi$ . Hence  $\Phi$  is originally satisfiable.  $\square$

Notice that the size of the automaton  $\mathcal{B}_\Phi$  is exponential in the length of  $\Phi$ , as it has potentially as many states as there are types for  $\text{cl}(\Phi^\downarrow)$ . Since checking universality of an NFA over a unary alphabet is coNP-complete (Stockmeyer and Meyer 1973), we obtain a coNExpTime upper bound for original satisfiability.

**Corollary 14.** *Original satisfiability of  $\text{TLD}^-$  formulas is in coNExpTime.*

As a last step in the study of satisfiability, we now consider the general case of a  $\text{TLD}^-$  formula such that the only occurrence of the formula  $\blacklozenge a$  may be satisfied at a later point in time. As mentioned already, if the formula is originally satisfiable then it is also satisfiable but the converse may not hold. Importantly, if  $\mathcal{P} \models \Phi$  then there exists a point in time  $n \in \mathbb{N}$  such that  $\mathcal{P}, n \models \blacklozenge a$  (see Lemma 8). This leads to the following characterisation of satisfiability.

**Theorem 15.**  *$\Phi$  is satisfiable iff there exists an  $n \in \mathbb{N}$  such that for every  $m \in \mathbb{N}$  there is an LTL interpretation  $I_m$  such that (i)  $I_m \models \Phi^\downarrow$ , (ii)  $I_m, n + m \models a$ , and (iii) for all  $k, 0 \leq k < m$ ,  $I_m, n + k \not\models a$ .*

Recall also that the GBA  $\mathcal{A}_\Phi$  accepts all the LTL models of  $\Phi^\downarrow$ . Following the ideas developed for original satisfiability, we can see that  $\Phi$  is satisfiable iff there is an  $n$  such that for every  $m$  there is a successful run  $\rho_m$  in  $\mathcal{A}_\Phi$  where  $\rho_m$  visits a state containing  $a$  after  $n + m$  transitions, and none of the states in the transitions between  $n$  and  $n + m - 1$  contain  $a$ . This motivates the following definition.

**Definition 16.** Let  $\mathcal{A}_\Phi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$  be the reduced GBA for the  $\text{TLD}^-$  formula  $\Phi$ . We define the sets  $\text{Ini}_i$  for  $i \in \mathbb{N}$  inductively as follows:  $\text{Ini}_0 := \text{Ini}$ ; and

$$\text{Ini}_{i+1} = \{s \in Q \mid (s', V, s) \in \Delta, s' \in \text{Ini}_i\}.$$

For each  $i \in \mathbb{N}$ , define the NBA  $\mathcal{B}_\Phi^i := (Q, \{1\}, \Delta', \text{Ini}_i, F)$ , where  $\Delta'$  and  $F$  are as in Definition 12.

Intuitively, each of the automata  $\mathcal{B}_\Phi^i$  corresponds to the origin automaton that checks that the conditions for satisfying  $\blacklozenge a$  can be satisfied, but rather than considering the initial states from  $\mathcal{A}_\Phi$ , it is initialized with those states reached after  $i$  transitions. Since  $\mathcal{A}_\Phi$  is reduced, from every state reached after  $i$  transitions it is still possible to complete a successful run, and hence a model of  $\Phi^\downarrow$ . Hence, we get the following.

**Theorem 17.** *The  $\text{TLD}^-$  formula  $\Phi$  is satisfiable iff there exists an  $n \in \mathbb{N}$  such that the NFA  $\mathcal{B}_\Phi^n$  accepts the language  $1^*$ .*

There are two important things to consider. First, the size of each of the automata  $\mathcal{B}_\Phi^i$  is exponential in the size of the formula  $\Phi$ . Second, since  $\text{Ini}_i \subseteq Q$  for all  $i \in \mathbb{N}$ , then in order to verify satisfiability, it suffices to consider  $n \leq 2^{|\mathcal{Q}|}$ . We thus obtain the following result.

**Theorem 18.** *Satisfiability of TLD<sup>-</sup> formulas is in NExpTime<sup>NP</sup>.*

*Proof.* We can guess in non-deterministic exponential time (in  $|\Phi|$ ) a number  $n$  smaller or equal to  $2^{|\mathcal{Q}|}$ , using a logarithmic encoding. Through a multiplication of the reachability matrix,  $\text{Ini}_n$  (and hence  $\mathcal{B}_\Phi^n$ ) is computed in exponential time. We can then call an NP oracle to check universality of this automaton.  $\square$

Interestingly, the algorithm described in this proof calls the NP oracle exactly once. Using the results from Hemachandra (1989), it follows that TLD<sup>-</sup> satisfiability is in the strong exponential hierarchy, which collapses (i.e., is equivalent) to P<sup>NE</sup>.

**Corollary 19.** *Satisfiability of TLD<sup>-</sup> formulas is in P<sup>NE</sup>.*

## Including Probabilities

In the previous section we focused on the problem of satisfiability of a TLD<sup>-</sup> formula, for which the only feature of the probabilistic constructor  $\blacklozenge$  needed is that the distribution  $\delta$  is complete. That property was used to justify the need for having an infinite class of LTL interpretations in every TLD model of a formula. However, our main interest is in fact to understand the uncertainty described through this constructor, and to deduce some consequences out of it.

Notice first that a TLD interpretation  $\mathcal{P} = (\mathcal{I}, \mu)$  defines, in fact, probability distributions for different LTL formulas to hold at each point in time.

**Definition 20.** Let  $\mathcal{P} = (\mathcal{I}, \mu)$  be a TLD interpretation,  $\varphi$  an LTL formula, and  $n \in \mathbb{N}$ . The *probability of satisfying  $\varphi$  at time  $n$  w.r.t.  $\mathcal{P}$*  is  $P_{\mathcal{P}}(\varphi, n) := \mu(\{I \in \mathcal{I} \mid I, n \models \varphi\})$ .

In general, we are interested in deducing properties of these probability distributions, when the class of TLD interpretations is restricted to the models of a given formula  $\Phi$ . The most basic such kinds of properties are the lower and upper probability bounds.

**Definition 21.** Let  $\Phi$  be a TLD formula,  $\varphi$  an LTL formula, and  $n \in \mathbb{N}$ . We define  $P_{\text{inf}}^\Phi(\varphi, n) := \inf_{\mathcal{P} \models \Phi} P_{\mathcal{P}}(\varphi, n)$  and  $P_{\text{sup}}^\Phi(\varphi, n) := \sup_{\mathcal{P} \models \Phi} P_{\mathcal{P}}(\varphi, n)$ .

These probabilistic bounds allow us to understand the likelihood that a given situation, defined by the LTL formula  $\varphi$  holds at time  $n$ . In the worst case, we would have that these probabilities are between 0 and 1, which is not informative, while in the best case,  $P_{\text{inf}}$  and  $P_{\text{sup}}$  coincide, which yields perfect information about the uncertainty of the formula of interest.

Notice that the problem of deciding whether  $P_{\text{inf}}$  ( $P_{\text{sup}}$ ) is smaller or equal (greater or equal, respectively) to a given constant  $\alpha \in [0, 1]$  is at least as hard as deciding satisfiability

---

### Algorithm 1 Computation of $\widehat{P}_{\text{inf}}^\Phi$ and $\widehat{P}_{\text{sup}}^\Phi$

---

```

 $P_{\text{inf}} \leftarrow 0$ 
 $P_{\text{sup}} \leftarrow 1$ 
for  $n \in \mathbb{N}$  do
  if  $1^n \notin L(\mathcal{B}_\Phi^+)$  then  $P_{\text{sup}} \leftarrow P_{\text{sup}} - \delta(n)$ 
  if  $1^n \notin L(\mathcal{B}_\Phi^-)$  then  $P_{\text{inf}} \leftarrow P_{\text{inf}} + \delta(n)$ 
return  $P_{\text{inf}}, P_{\text{sup}}$ 

```

---

of TLD formulas. Indeed, the formula  $\Phi$  is unsatisfiable iff  $P_{\text{inf}}^\Phi(\neg\top, 0) = 1$  iff  $P_{\text{sup}}^\Phi(\top, 0) = 0$ <sup>3</sup>.

As we did in the previous section, we will study the problem of computing these infima and suprema in parts. First, let us assume w.l.o.g. that we are interested in finding the probability of observing a propositional variable  $b$  that appears in the TLD formula  $\Phi$ ; if we are interested in a more complex formula  $\psi$ , we can simply conjoin  $\Phi$  with the formula  $\square((b \wedge \psi) \vee (\neg b \wedge \neg \psi))$ , where  $b$  is a fresh variable not appearing in  $\Phi$ . Moreover, we consider only the initial point in time  $n = 0$ . That is, we focus on computing  $P_{\text{inf}}^\Phi(b, 0)$  and  $P_{\text{sup}}^\Phi(b, 0)$ . Finally, we assume for simplicity that  $\Phi$  is satisfiable; otherwise, the bounds are already known, as described before.

For now, we look only at models where  $\mathcal{P}, 0 \models \blacklozenge a$ ; that is, we restrict our attention to models that witness original satisfiability. In other words, we are at the moment interested in computing  $\widehat{P}_{\text{inf}}^\Phi := \inf_{\mathcal{P}, 0 \models \Phi \wedge \blacklozenge a} P_{\mathcal{P}}(b, 0)$  and  $\widehat{P}_{\text{sup}}^\Phi := \sup_{\mathcal{P}, 0 \models \Phi \wedge \blacklozenge a} P_{\mathcal{P}}(b, 0)$ . We will see how to deal with the general case later. As seen in the previous section, it suffices to consider such models containing LTL interpretations  $I_n \models \Phi^\downarrow$  for each  $n \in \mathbb{N}$  such that  $I_n, n \models a$  and  $I_n, k \not\models a$  for all  $0 \leq k < n$ . We are now interested in finding out which of these models also satisfy  $b$  at time 0.

Recall from the previous section that the automaton  $\mathcal{B}_\Phi$  is such that, for every  $n \in \mathbb{N}$ ,  $\mathcal{B}_\Phi$  accepts the word  $1^n$  iff there is a model of  $\Phi^\downarrow$  such that  $a$  is true at the  $n$ -th point in time, and nowhere before. We now refine this idea by considering, in addition, whether those models satisfy  $b$  or not.

**Definition 22.** Let  $\mathcal{B}_\Phi = (Q, \{1\}, \Delta, \text{Ini}, F)$  be the origin automaton for the TLD<sup>-</sup> formula  $\Phi$ . We define the two NFA  $\mathcal{B}_\Phi^+ = (Q, \{1\}, \Delta, \text{Ini}^+, F)$  and  $\mathcal{B}_\Phi^- = (Q, \{1\}, \Delta, \text{Ini}^-, F)$ , with  $\text{Ini}^+ = \{s \in \text{Ini} \mid b \in s\}$ , and  $\text{Ini}^- = \{s \in \text{Ini} \mid b \notin s\}$ .

Intuitively,  $\mathcal{B}_\Phi^+$  detects the values  $n \in \mathbb{N}$  such that it is possible to construct a model  $I_n \models b \wedge \Phi^\downarrow$  where  $I_n, n \models a$ , and  $I_n, k \not\models a$  for all  $0 \leq k < n$ . Similarly, for  $\mathcal{B}_\Phi^-$  but with  $I_n \models \neg b \wedge \Phi^\downarrow$ . This is the idea behind Algorithm 1.

Strictly speaking, the procedure from Algorithm 1 does not describe an algorithm as the **for** loop cycles over all natural numbers. Notice, however, that each iteration can be computed in linear time in the size of the automaton  $\mathcal{B}_\Phi$ : to check whether the automata accepts  $1^0 = \varepsilon$ , one simply needs to see whether the initial and final states are not disjoint. One can then compute all the states that are reach-

---

<sup>3</sup>We use the convention that an infimum over an empty set is the maximum possible value, and dually for suprema.

able in one transition, which allows the next iteration to be treated similarly. Thus, this describes an *any-time* algorithm that provides better and better approximations to  $P_{\text{inf}}$  and  $P_{\text{sup}}$  as time progresses.

Notice, moreover, that since  $\delta$  is a probability distribution,  $\sum_{n \in \mathbb{N}} \delta(n) = 1$ , and hence for any given  $\epsilon > 0$  there is an  $n_\epsilon \in \mathbb{N}$  such that  $\sum_{n \geq n_\epsilon} \delta(n) < \epsilon$ . In other words, given a desired level of approximation  $\epsilon$ , it is possible to halt the execution of the process after  $n_\epsilon$  steps with the guarantee that the computed  $P_{\text{inf}}$  and  $P_{\text{sup}}$  differ from the real  $\widehat{P}_{\text{inf}}^\Phi$  and  $\widehat{P}_{\text{sup}}^\Phi$  by at most  $\epsilon$ .

Throughout the **for** loop, the procedure keeps the following invariant: for every model  $\mathcal{P}$  of  $\Phi \wedge \Diamond a$ , it holds that  $P_{\text{inf}} \leq P_{\mathcal{P}}(b, 0) \leq P_{\text{sup}}$ . Furthermore, if  $\Phi$  is satisfiable, the procedure suggests the construction of two models  $\mathcal{P}_1$  and  $\mathcal{P}_2$  such that  $P_{\text{inf}} = P_{\mathcal{P}_1}(b, 0)$  and  $P_{\text{sup}} = P_{\mathcal{P}_2}(b, 0)$ . This means that the infima and suprema in this case are both witnessed as minima and maxima. In addition, the differences between these two bounds are observed when it is possible to find LTL models  $I_n, I'_n$  both satisfying  $a$  first at point  $n$ , such that  $I_n \models b$  and  $I'_n \not\models b$ . Hence we have the following result.

**Theorem 23.** *For every value  $x$ ,  $\widehat{P}_{\text{inf}}^\Phi \leq x \leq \widehat{P}_{\text{sup}}^\Phi$ , there exists a model  $\mathcal{P}$  of  $\Phi$  such that  $P_{\mathcal{P}}(b, 0) = x$ .*

*Proof.* If  $\widehat{P}_{\text{inf}}^\Phi \leq x \leq \widehat{P}_{\text{sup}}^\Phi$ , then  $x$  can be rewritten as

$$x = \alpha \widehat{P}_{\text{inf}}^\Phi + (1 - \alpha) \widehat{P}_{\text{sup}}^\Phi$$

for some constant  $\alpha \in [0, 1]$ . We build a model  $\mathcal{P}$  of  $\Phi$  as follows. For every  $n \in \mathbb{N}$ , if  $1^n \notin L(\mathcal{B}_\Phi^+)$ , then take a model  $I_n$  such that  $I_n, 0 \models \neg b$  and  $I_n$  satisfies  $a$  first at time  $n$ , and define  $\mu(I_n) := \delta(n)$ . Similarly, if  $1^n \notin L(\mathcal{B}_\Phi^-)$  take  $I_n, 0 \models b$ , with  $\mu(I_n) := \delta(n)$ . Otherwise, there are models  $I_n, I'_n$  s.t.  $I_n \models b$  and  $I'_n \models \neg b$ ; define  $\mu(I_n) := \alpha \delta(n)$  and  $\mu(I'_n) := (1 - \alpha) \delta(n)$ . This defines a model  $\mathcal{P}$  of  $\Phi$  and  $P_{\mathcal{P}}(b, 0) = x$ .  $\square$

Notice that the important feature of Algorithm 1 is to find out the class of words that are accepted by  $\mathcal{B}_\Phi^+$  and  $\mathcal{B}_\Phi^-$ . In fact, from the arguments presented in this section, we can conclude the following.

**Corollary 24.** *Let  $N^+ := \{n \in \mathbb{N} \mid 1^n \in L(\mathcal{B}_\Phi^+)\}$  and  $N^- := \{n \in \mathbb{N} \mid 1^n \in L(\mathcal{B}_\Phi^-)\}$ . Then,*

$$\begin{aligned} \widehat{P}_{\text{sup}}^\Phi &= \sum_{n \in N^+} \delta(n), & \text{and} \\ \widehat{P}_{\text{inf}}^\Phi &= 1 - \sum_{n \in N^-} \delta(n). \end{aligned}$$

This result shows that, in order to find the lower and upper bounds for the probability of observing the variable  $b$  at time 0, it suffices to understand the languages accepted by the two automata  $\mathcal{B}_\Phi^+$  and  $\mathcal{B}_\Phi^-$ , provided that we know the behaviour of the distribution  $\delta$ . Notice that these two automata are unary; that is, they read only one alphabet symbol. Thus, they can be transformed to equivalent automata in *Chrobak normal form* (Chrobak 1986;

To 2008), from which the language accepted by them can be easily read in terms of linear binomials.

More precisely, the results by Chrobak, which were later improved and corrected in (Martinez 2002; To 2008) show that, given a unary non-deterministic finite automaton  $\mathcal{A}$  having  $n$  states, it is possible to effectively construct a polynomial number of arithmetic progressions of the form  $\alpha_i + \beta_i \mathbb{N}$ ,  $1 \leq i \leq k$ , with  $k \in \mathcal{O}(n^2)$ , such that

$$L(\mathcal{A}) = \{1^{\alpha_i + \beta_i \ell} \mid 1 \leq i \leq k, \ell \in \mathbb{N}\}.$$

Moreover, each of the constants  $\alpha_i, \beta_i$ ,  $1 \leq i \leq k$  is bounded quadratically by  $n$ , too. Applying this idea to the automata  $\mathcal{B}_\Phi^+$  and  $\mathcal{B}_\Phi^-$ , it is possible to effectively compute the sets  $N^+$  and  $N^-$  and, by extension, also the probabilistic bounds  $\widehat{P}_{\text{inf}}^\Phi$  and  $\widehat{P}_{\text{sup}}^\Phi$ ; for instance, by following the methods devised by Matos (1994) for computing over periodic sets of integers.

We have limited so far our analysis to the values  $\widehat{P}_{\text{inf}}^\Phi$  and  $\widehat{P}_{\text{sup}}^\Phi$ , where the  $\Diamond$  constructor is satisfied at the first point in time. In general, however, there might be additional models that do not satisfy this constraint. That is, we know that  $P_{\text{inf}}^\Phi(b, 0) \leq \widehat{P}_{\text{inf}}^\Phi$  and  $P_{\text{sup}}^\Phi(b, 0) \geq \widehat{P}_{\text{sup}}^\Phi$ , but these inequalities may be strict.

*Example 25.* Consider the formula  $\Phi_2 := \Diamond a \wedge (\neg a \vee \neg b)$ , where  $\delta$  is the geometric distribution with parameter 0.5. It is easy to see that any model  $\mathcal{P} = (\mathcal{I}, \mu)$  of  $\Phi_2$  with  $\mathcal{P}, 0 \models \Diamond a$  is such that for every  $I \in \mathcal{I}$ , if  $I, 0 \models a$ , then  $I, 0 \not\models b$ . Thus,  $\widehat{P}_{\text{sup}}^{\Phi_2} = 0.5$ . However, simply by delaying the satisfaction of  $\Diamond a$  to the successive point in time, it is possible to satisfy  $b$  at the initial point in time. Hence  $P_{\text{sup}}^{\Phi_2}(b, 0) = 1$ . A similar example can be built to show that it is possible to have formulas where  $P_{\text{inf}}^\Phi(b, 0) < \widehat{P}_{\text{inf}}^\Phi$ .

To find the precise values of  $P_{\text{inf}}^\Phi(b, 0)$  and  $P_{\text{sup}}^\Phi(b, 0)$ , we combine Algorithm 1—or, more precisely, the finite process obtained through the Chrobak normal form of the automata  $\mathcal{B}_\Phi^+$  and  $\mathcal{B}_\Phi^-$ —with the ideas behind Theorems 17 and 18. Formally, let  $\mathcal{A}_\Phi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$  be the reduced GBA for  $\Phi$ . We inductively define  $\text{Ini}_0^+ := \{s \in \text{Ini} \mid b \in s\}$ ,  $\text{Ini}_0^- := \{s \in \text{Ini} \mid b \notin s\}$ , and for every  $n \in \mathbb{N}$

$$\text{Ini}_{i+1}^+ := \{s \in Q \mid (s', V, s) \in \Delta, s' \in \text{Ini}_i^+\}$$

$$\text{Ini}_{i+1}^- := \{s \in Q \mid (s', V, s) \in \Delta, s' \in \text{Ini}_i^-\}.$$

These define the NBAs  $\mathcal{B}_\Phi^{i+} = (Q, \{1\}, \Delta', \text{Ini}_i^+, F)$  and  $\mathcal{B}_\Phi^{i-} = (Q, \{1\}, \Delta', \text{Ini}_i^-, F)$ , where  $\Delta'$  and  $F$  are as in Definition 12 of the origin automaton.

Let  $M$  be the set of all  $n \in \mathbb{N}$  such that  $\mathcal{B}_\Phi^n$  is universal; that is, all points in time where it is possible to satisfy  $\Diamond a$ . For each  $n \in M$ , let  $\widehat{P}_{\text{inf}}^n, \widehat{P}_{\text{sup}}^n$  be the result of executing Algorithm 1 but replacing  $\mathcal{B}_\Phi^+$  and  $\mathcal{B}_\Phi^-$  with  $\mathcal{B}_\Phi^{n+}$  and  $\mathcal{B}_\Phi^{n-}$ , respectively. Then, it follows that  $P_{\text{inf}}^\Phi(b, 0) = \inf_{n \in M} \widehat{P}_{\text{inf}}^n$  and  $P_{\text{sup}}^\Phi(b, 0) = \sup_{n \in M} \widehat{P}_{\text{sup}}^n$ .

As argued already, each of the local bounds  $\widehat{P}_{\text{inf}}^n$  and  $\widehat{P}_{\text{sup}}^n$  can be effectively computed by means of the Chrobak normal form of the underlying automata. Moreover, since  $\mathcal{B}_\Phi$

has finitely many states, it suffices to look into a finite number of automata  $\mathcal{B}_\Phi^{n\pm}$  (bounded exponentially by the number of states of  $\mathcal{B}_\Phi$ ) before these automata are equivalent to previously observed cases. As a consequence, the infima and suprema of the probabilities of observing  $b$  at the first point in time in a model of  $\Phi$  can be effectively computed; moreover, these bounds are reachable.

**Theorem 26.** *There exists models  $\mathcal{P}, \mathcal{P}'$  of  $\Phi$  such that  $P_{\mathcal{P}}(b, 0) = P_{\inf}^\Phi(b, 0)$  and  $P_{\mathcal{P}'}(b, 0) = P_{\sup}^\Phi(b, 0)$ .*

We have already solved the problem of computing the probability of observing  $b$  at the first point in time, but in general, we would like to compute  $P_{\inf}^\Phi(\varphi, n)$  and  $P_{\sup}^\Phi(\varphi, n)$  for an LTL formula  $\varphi$  and  $n \in \mathbb{N}$ . Let  $b$  a propositional variable not appearing in  $\Phi$ . Then, we can define the new TLD<sup>-</sup> formula  $\Psi := \Phi \wedge ((b \wedge \bigcirc^n \varphi) \vee (\neg b \wedge \neg \bigcirc^n \varphi))$ . It is easy to see that for every model  $\mathcal{P} = (\mathcal{I}, \mu)$  of  $\Phi$  there is a model  $\mathcal{P}'$  of  $\Psi$  such that  $P_{\mathcal{P}}(\varphi, n) = P_{\mathcal{P}'}(b, 0)$ : simply extend every interpretation  $I \in \mathcal{I}$  such that  $I, n \models \varphi$  to satisfy also the fresh variable  $b$  at time 0. Conversely, every model  $\mathcal{P}$  of  $\Psi$  is such that  $P_{\mathcal{P}}(\varphi, n) = P_{\mathcal{P}}(b, 0)$ . Hence, we obtain the following result.

**Theorem 27.** *Given the TLD<sup>-</sup> formula  $\Phi$ , the LTL formula  $\varphi$ , and  $n \in \mathbb{N}$ , let  $\Psi := \Phi \wedge ((b \wedge \bigcirc^n \varphi) \vee (\neg b \wedge \neg \bigcirc^n \varphi))$ , where  $b$  is a fresh propositional variable not appearing in  $\Phi$ . Then  $P_{\inf}^\Psi(b, 0) = P_{\inf}^\Phi(\varphi, n)$  and  $P_{\sup}^\Psi(b, 0) = P_{\sup}^\Phi(\varphi, n)$ .*

## Expectations

Throughout the previous section, we focused on the problem of approximating the probability of observing a given event  $\varphi$  at specific points in time, under the uncertainty provided by the  $\blacklozenge$  constructor. Recall that the motivation for introducing this new constructor was to provide more fine-grained information about *when* might an eventuality be resolved. In the same spirit, we may want to find the expected time until an additional event  $\varphi$  is observed.

**Definition 28.** Let  $\mathcal{P} = (\mathcal{I}, \mu)$  be a TLD interpretation, and  $\varphi$  an LTL formula. The *expected time to observe  $\varphi$*  w.r.t.  $\mathcal{P}$  is  $E_{\mathcal{P}}(\varphi) := \sum_{n=0}^{\infty} n \cdot \mu(\mathcal{I}_n^\varphi \setminus \bigcup_{j=0}^{n-1} \mathcal{I}_j^\varphi)$ .<sup>4</sup>

Given a TLD<sup>-</sup> formula  $\Phi$ , the *expected time to observe  $\varphi$*  w.r.t.  $\Phi$  is  $E_\Phi(\varphi) := \inf_{\mathcal{P} \models \Phi} E_{\mathcal{P}}(\varphi)$ .

In essence, every model  $\mathcal{P}$  of  $\Phi$  describes a probabilistic setting, in which the expected number of steps until the formula  $\varphi$  is observed is well-defined. The value of  $E_\Phi(\varphi)$  is then an *optimistic* view on this expectation, as it refers to the earliest possible observation of  $\varphi$  within models of  $\Phi$ . Notice that we could have also defined a *pessimistic* variant substituting the infimum by a supremum; however, that expectation will often diverge, providing no relevant information about the formula of interest.

As mentioned,  $E_\Phi(\varphi)$  intuitively refers to the model of  $\Phi$  where  $\varphi$  is observed as early as possible. Indeed, if  $P_{\sup}^\Phi(\varphi, 0) = 1$ , then  $E_\Phi(\varphi) = 0$ ; that is, there is an optimistic assessment that  $\varphi$  will be observed at time 0. One could try to generalize this idea to find the expectation through the values of  $P_{\sup}^\Phi([\varphi]^n, 0)$ , where for every

<sup>4</sup>Recall that  $\mathcal{I}_n^\varphi$  was introduced in Definition 1.

$n \in \mathbb{N}$   $[\varphi]^n := \bigwedge_{i=0}^{n-1} \bigcirc^i \neg \varphi \wedge \bigcirc^n \varphi$ . However, this idea does not work. Consider the formula  $\Phi_3 := \blacklozenge a \wedge \square(\neg a \vee \neg b)$ . Then for every  $n \in \mathbb{N}$ ,  $P_{\sup}^{\Phi_3}([b]^n, 0) = 1 - 1/2^{n+1}$ , but  $E_{\Phi_3}(b) = \frac{1}{2}$ . The reason for this mismatch is that the suprema found for each  $n$  are reached by different models, while the expectation looks at one model only. Still, it is possible to adapt the automata-based techniques to handle this case as well.

For simplicity, we focus only on finding the expectation for TLD models  $\mathcal{P}$  such that  $\mathcal{P}, 0 \models \blacklozenge a$ . The results can be extended to the general case as we have done in the previous sections. Let  $\mathcal{A}_\Phi = (Q, \Sigma, \Delta, \text{Ini}, \mathcal{F})$  be the reduced GBA for  $\Phi$ . For every propositional variable  $x$  appearing in  $\Phi^\downarrow$ , let  $Q_x := \{s \in Q \mid x \in s\}$  be the set of all states of  $\mathcal{A}_\Phi$  that contain  $x$ . For each  $n \in \mathbb{N}$ , we define

$$\text{Ini}'_n := \{s_0 \in \text{Ini} \mid \exists s_1, \dots, s_n \in Q \text{ such that } s_n \in Q_b \text{ and } \forall i < n. (s_i, V, s_{i+1}) \in \Delta, s_i \notin Q_b\}.$$

That is,  $\text{Ini}'_n$  contains all the initial states from  $\mathcal{A}_\Phi$  that allow for runs where the first  $n - 1$  states do not contain  $b$ , but the  $n$ -th state contains  $b$ . In other words, these are the initial states for accepting interpretations where  $b$  is satisfied for the first time at time  $n$ .

For each  $i \in \mathbb{N}$ , we define now the automaton

$$\mathcal{C}_\Phi^i := (Q, \{1\}, \Delta', \text{Ini}'_i, F),$$

where  $\Delta'$  and  $F$  are as in Definition 12. Notice that  $\mathcal{C}_\Phi^0$  and  $\mathcal{B}_\Phi^+$  are equivalent, and hence  $\mathcal{C}_\Phi^0$  detects all LTL models of  $\Phi^\downarrow$  where  $b$  can be satisfied at the initial time. Likewise,  $\mathcal{C}_\Phi^i$  detects the models where  $b$  is observed first at time  $i$ ; however, due to non-determinism, it may also accept other models. Moreover, it may also include interpretations where it is also possible to satisfy  $b$  earlier than time  $i$ , as observed by some  $\mathcal{C}_\Phi^j$ ,  $j < i$ . To handle the former issue, we define NFA  $\mathcal{A}_\Phi^i$  that accepts all the prefixes of  $L(\mathcal{A}_\Phi)$  observing the variable  $b$  at time  $i$  exactly. The latter issue is handled through the new NFA

$$\mathcal{D}_\Phi^i := \mathcal{C}_\Phi^i \cap \mathcal{A}_\Phi^i \cap \bigcap_{j=0}^{i-1} \overline{\mathcal{C}_\Phi^j \cap \mathcal{A}_\Phi^j},$$

where the intersection and complementation refer to the standard operations over NFA, which can be effectively computed (Rich 2007).

It then follows that the automata  $\mathcal{D}_\Phi^i$  accept the unary words that describe models of  $\Phi^\downarrow$  where the earliest possible that  $b$  is satisfied is at time  $i$ . Since the lengths of these words refer to the point in time when  $a$  will be observed for the first time—as all the automata are based on the construction of the origin automaton—they collectively express the probability of observing  $b$  for the first time at point  $i$ , in a model of  $\Phi^\downarrow$ . Hence, we can use them to compute the expectation. For every  $n \in \mathbb{N}$ , let  $N^n := \{m \in \mathbb{N} \mid 1^m \in L(\mathcal{D}_\Phi^n)\}$  and  $P(n) = \sum_{i \in N^n} \delta(i)$ , then we have the following result.

**Theorem 29.**  $E_\Phi(b) = \sum_{n \in \mathbb{N}} n \cdot P(n)$ .

Notice that each of the summands  $P(n)$  can be effectively computed, as it requires only the construction of a



non-deterministic finite automaton, and the computation of its Chrobak normal form. However, the computation of the expectation itself, as described by Theorem 29 is not effective. In fact, it requires the construction of infinitely many automata  $\mathcal{A}_\Phi^i$  which are all different. Still, the theorem allows us to conclude that the expectation is always witnessed by a model.

**Corollary 30.** *There exists a model  $\mathcal{P}$  of  $\Phi$  such that  $E_\Phi(\varphi) = E_{\mathcal{P}}(\varphi)$ .*

In addition, similarly to Algorithm 1, the process suggested by Theorem 29 can be transformed into an any-time algorithm that computes increasingly better approximations of the expectation, and can be halted when the differences between results are below an accepted margin of error.

## Related Work

Before concluding this paper, it is worth noting that many probabilistic extensions of temporal logics have been proposed over the years. For a relevant, although slightly outdated overview of these logics, we refer the interested reader to (Konur 2010). Here we briefly discuss some of the more relevant or newer approaches.

The most basic probabilistic extension of LTL is perhaps pLTL (Morão 2011). This logic extends basic probabilistic propositional logic (Nilsson 1986) with the temporal operators of LTL. Thus, every point in time represents an uncertain scenario, which is represented by a set of (propositional) interpretations. This differs greatly from the semantics of TLD, where a *world* corresponds to a potential full evolution of the system (i.e., an LTL interpretation). The biggest difference between pLTL and TLD is that the former can characterise only distributions that have finite memory. A simple example that cannot be captured by pLTL is the formula  $\Diamond_\delta \phi$ , where  $\delta$  is the Pascal distribution. Since this distribution depends on the time that one has waited already, the only way to model it in pLTL is to explicitly provide the probability for each of the (infinitely many) timepoints.

Another extension of LTL that has been studied in more detail is PLTL (Ognjanovic 2006), which provides a rich syntax for expressing evolving probabilities. The semantics proposed in (Ognjanovic 2006) are closer to ours in that they correspond to measurable spaces of LTL interpretations. However, this logic does not provide any way to represent and handle complex distributions over time, as we do for TLD.

There exist, as well, other alternatives for representing complex evolution of uncertainty over time. For instance, Ognjanovic et al. (2012) propose a method for handling the evidence provided by observations made during time, in such a way that the beliefs over uncertain statements are updated to agree with the evidence. In a different direction, Doder and Ognjanovic (2015) provide comparison constructors that can be used to express, e.g., that the probability of a washing machine breaking now is smaller than that of it breaking next year, without having to express these probabilities explicitly.

One important application that combines probabilities and temporal formulas is probabilistic model checking (PMC)

(Forejt et al. 2011; Ding et al. 2011). In PMC, one tries to verify that the probability of observing a property over a probabilistic transition system (often a Markov chain) is greater or equal to some pre-specified bound. Among many advances in this area, techniques for studying limit behaviours and formula quantiles have been developed (Klein et al. 2018). For some types of probability distributions, such as the geometric, it is easy to describe a Markov chain that will generate them. However, verifying a property over such a Markov chain (or making probabilistic entailments over this environment) is not the same as dealing with our  $\Diamond$  formulas, which may also refer to very complex distributions.

Finally, without describing them in detail, we acknowledge the existence of probabilistic temporal logics for continuous (Tiger and Heintz 2016) and branching (Hart and Sharir 1984; Forejt et al. 2011) time, for counting the frequency of occurrences of a property (Bollig, Decker, and Leucker 2012), and for dealing with complex actions (Paleo 2016), among many others.

## Conclusions

We have introduced the new logic TLD, which extends LTL with the possibility of expressing uncertain knowledge about the waiting time required until an eventuality is satisfied. Thus, TLD refines the knowledge that is expressed by the diamond constructor in LTL.

Since the full TLD is capable of expressing properties that can only be satisfied in uncountable models, we focused our attention to the sublogic  $\text{TLD}^-$ , where the probabilistic constructor  $\Diamond$  needs to be satisfied at most once. Moreover, we considered the distribution  $\delta$  to be complete; i.e., that it assigns a positive probability of observing the target formula to every point in time. Under these conditions, we have studied the complexity of deciding satisfiability of  $\text{TLD}^-$  formulas, and described methods for computing probabilistic entailments as well as expectation statements. To obtain our results, we leveraged known results from the area of LTL, and in particular, the existence of automata that accept all models of an LTL formula. The probability computations were based on the construction of the Chrobak normal form for unary automata and combinations of regular expressions and integer polynomials.

As future work, we intend to strengthen our results for  $\text{TLD}^-$  and to extend them to more expressive variants of TLD. In the former context, we will improve the complexity bounds for all the problems that we have presented, and study the potential of using weighted automata for dealing with the probabilistic values directly. For the latter, we will weaken the syntactic constraints that define  $\text{TLD}^-$ . It should be relatively straightforward in theory—although requiring additional algebraic machinery—to extend our results to arbitrary formulas as long as the  $\Diamond$  operator is satisfied in only finitely many points in time, in contrast to only one as in  $\text{TLD}^-$ . As a simple example, the disjunction of several  $\Diamond$  formulas should cause no additional problems. In addition, we will consider different properties for the probability distributions  $\delta$  under which reasoning can still be effective.

If infinitely many  $\Diamond$  properties need to be satisfied, then in general we cannot avoid considering uncountable models,

as shown in Theorem 4. One potential way to describe these models, which are composed by uncountably many LTL interpretations, is to represent them as infinite trees as depicted in Figure 1. One could then try to use *tree* automata to verify satisfiability and derive probabilistic and expectation entailments, perhaps also with the help of weighted transitions. We intend to further explore this possibility.

## References

- Billingsley, P. 1995. *Probability and Measure*. John Wiley and Sons, third edition.
- Bollig, B.; Decker, N.; and Leucker, M. 2012. Frequency linear-time temporal logic. In *Sixth International Symposium on Theoretical Aspects of Software Engineering, TASE 2012*, 85–92.
- Chomicki, J. 1994. Temporal query languages: A survey. In Gabbay, D. M., and Ohlbach, H. J., eds., *Temporal Logic*, 506–534. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Chrobak, M. 1986. Finite automata and unary languages. *Theor. Comput. Sci.* 47(3):149–158.
- Clarke, E. M.; Emerson, E. A.; and Sistla, A. P. 1986. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.* 8(2):244–263.
- Ding, X. C.; Smith, S. L.; Belta, C.; and Rus, D. 2011. LTL control in uncertain environments with probabilistic satisfaction guarantees. *CoRR* abs/1104.1159.
- Doder, D., and Ognjanovic, Z. 2015. A probabilistic logic for reasoning about uncertain temporal information. In *Proc. of the 31st Conference on Uncertainty in Artif. Intell., UAI 2015*, 248–257.
- Emerson, E. A., and Srinivasan, J. 1989. Branching time temporal logic. In de Bakker, J. W.; de Roever, W. P.; and Rozenberg, G., eds., *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, 123–172. Berlin, Heidelberg: Springer.
- Forejt, V.; Kwiatkowska, M.; Norman, G.; and Parker, D. 2011. Automated verification techniques for probabilistic systems. In *Form. Methods for Eternal Networked Software Syst. (SFM’11)*, volume 6659 of *LNCS*, 53–113. Springer.
- Hart, S., and Sharir, M. 1984. Probabilistic temporal logics for finite and bounded models. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, 1–13.
- Hemachandra, L. A. 1989. The strong exponential hierarchy collapses. *J. of Comp. and System Sciences* 39(3):299 – 322.
- Kesten, Y.; Manna, Z.; McGuire, H.; and Pnueli, A. 1993. A decision algorithm for full propositional temporal logic. In Courcoubetis, C., ed., *Computer Aided Verification*, 97–109. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Klein, J.; Baier, C.; Chrszon, P.; Daum, M.; Dubslaff, C.; Klüppelholz, S.; Märcker, S.; and Müller, D. 2018. Advances in probabilistic model checking with PRISM: variable reordering, quantiles and weak deterministic büchi automata. *STTT* 20(2):179–194.
- Konur, S. 2010. Real-time and probabilistic temporal logics: An overview. *CoRR* abs/1005.3200.
- Koymans, R. 1990. Specifying real-time properties with metric temporal logic. *Real-Time Systems* 2(4):255–299.
- Lutz, C.; Wolter, F.; and Zakharyashev, M. 2008. Temporal description logics: A survey. In *15th International Symposium on Temporal Representation and Reasoning*, 3–14.
- Maler, O.; Nickovic, D.; and Pnueli, A. 2005. Real time temporal logic: Past, present, future. In Pettersson, P., and Yi, W., eds., *Formal Modeling and Analysis of Timed Systems*, 2–16. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Markey, N. 2003. Temporal Logic with Past is Exponentially More Succinct. *EATCS Bulletin* 79:122–128.
- Markey, N. 2004. Past is for free: on the complexity of verifying linear temporal properties with past. *Acta Informatica* 40(6):431–458.
- Martinez, A. 2002. Efficient computation of regular expressions from unary NFAs. In *4th Int. Workshop on Descriptive Complexity of Formal Systems - DCFs 2002*, volume Report No. 586, 174–187. Department of Computer Science, The University of Western Ontario, Canada.
- Matos, A. B. 1994. Periodic sets of integers. *Theoretical Computer Science* 127(2):287 – 312.
- Morão, L. 2011. Probabilistic distributed temporal logic. Master’s thesis, Universidade Técnica de Lisboa, Portugal.
- Nilsson, N. J. 1986. Probabilistic logic. *Artificial Intelligence* 28(1):71–88.
- Ognjanovic, Z.; Markovic, Z.; Raskovic, M.; Doder, D.; and Perovic, A. 2012. A propositional probabilistic logic with discrete linear time for reasoning about evidence. *Ann. Math. Artif. Intell.* 65(2-3):217–243.
- Ognjanovic, Z. 2006. Discrete linear-time probabilistic logics: Completeness, decidability and complexity. *J. Log. Comput.* 16(2):257–285.
- Paleo, B. W. 2016. An expressive probabilistic temporal logic. *CoRR* abs/1603.07453.
- Pnueli, A. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, 46–57.
- Pnueli, A. 1986. *Applications of temporal logic to the specification and verification of reactive systems: A survey of current trends*. Berlin, Heidelberg: Springer. 510–584.
- Rich, E. A. 2007. *Automata, Computability and Complexity: Theory and Applications*. Pearson Higher Ed., 1st edition.
- Stockmeyer, L. J., and Meyer, A. R. 1973. Word problems requiring exponential time (preliminary report). In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, STOC ’73, 1–9. New York, NY, USA: ACM.
- Tiger, M., and Heintz, F. 2016. Stream reasoning using temporal logic and predictive probabilistic state models. In *23rd Int. Symp. on Temp. Rep. and Reas., TIME’16*, 196–205.
- To, A. W. 2008. Unary finite automata vs. arithmetic progressions. *CoRR* abs/0812.1291.
- Vardi, M. Y., and Wolper, P. 1994. Reasoning about infinite computations. *Information and Computation* 115(1):1 – 37.