Possibilistic Reasoning on Finite-time Linear Temporal Logic

Virginia Carugno and Rafael Peñaloza

University of Milano-Bicocca, Milan, Italy
ORCID (Rafael Peñaloza): https://orcid.org/0000-0002-2693-5790

Abstract. Linear temporal logic on finite time (LTL_f) has been successfully employed as a declarative language to represent and reason about business processes. However, in their classical form they are unable to handle the uncertainty inherent to many application domains and model learning methods. To alleviate this, we propose a possibilistic extension of LTL_f designed to deal with qualitative uncertainty, first in a restricted form where uncertainty refers to full formulas, and then in a more expressive formalism with possibility as a full formula constructor. We present effective automata-based reasoning methods for this new language and show that it behaves computationally better than recently proposed probabilistic temporal logics.

1 Introduction

Linear temporal logic (LTL) [3, 28] is one of the main formalisms used for describing temporal and process specifications and behaviour. Its variant LTL_f [9], which restricts timelines to a finite execution has recently gained attention from the community dealing with declarative process modelling as, in contrast to verification or model checking where executions are assumed to run forever, business processes are considered to have an end point. In particular, LTL_f is the formalism underlying the declarative process modelling language Declare [27].

Uncertainty is an unavoidable feature of real-world domains, in particular when dealing with industrial applications and temporal executions. When an industrial process is set in motion, there are potential variations on the output which may be caused by limited knowledge (e.g., is the raw material of adequate quality?; has it been stored properly?) or by the randomness of the world (e.g., a shipment container may be lost at a sea storm, or an electronic device may suffer from a bit-flip). Classical logics are unable to characterise this uncertainty and, if used improperly—for instance, by considering unlikely scenarios as impossible—may lead to erroneous specifications and conclusions. Hence the need to extend LTL_f to handle uncertainty.

Probability theory [12] provides an effective and well-understood approach to deal with uncertainty in a quantitative manner. Indeed, many probabilistic temporal logics (typically over infinite time) have been proposed over the years [16, 10, 26, 25, 17]. A different strategy is to use Markov chains [24], like in PCTL [14] which associates to each state of a Markov chain a propositional valuation expressing the observable behaviour of the system at that state. To our knowledge, finite-time variants of PCTL have not been studied; however, the problem of finding out whether a PCTL formula has a finite model was shown to be undecidable [6]. The closest work to ours is PLTL_f [23] and its restriction PLTL_f^0, which was proposed as the backbone of ProbDeclare [22].

Probabilities, while technically well-understood, are cognitively and computationally very expensive. On the one hand, it has been observed that humans are not well adapted to reason probabilistically, making the construction of a probabilistic specification from expert knowledge much harder. On the other hand, adequate probabilistic reasoning requires knowledge about joint distributions and independence between events, which is not easy to obtain, and hard to manipulate numerically once it is available.

Possibility theory [11] provides an alternative, more *qualitative* method for managing uncertainty. In their most typical form, possibility measures simply consider the maximum possibility of the outcomes in an event, thus making the computation much simpler. In addition, the actual *values* of the possibility degrees take a second place behind their *order*—hence the idea that this measure is more qualitative than quantitative. This is helpful from the modelling point of view because variations in possibility degrees do not affect the overall results, as long as their ordering remains unchanged. Thus, a modeller only needs to specify what is more (or less) possible between two events, without worrying about assigning a precise value.

Possibilistic (infinite time) temporal logics have been studied in the context of model checking [18, 19], but to the best of our knowledge, never for finite-time temporal logics or for logical reasoning. In this paper, we cover that gap by proposing two possibilistic extensions of LTL_f. After a few preliminaries needed to understand the details of the paper, we first present the very simple ΠLTL_f^0, where possibilistic statements can only be made to classical LTL_f formulas. We show that deciding consistency of ΠLTL_f^0 KBs is computationally as expensive as for classical LTL_f, and present methods for finding inconsistency degrees and decide more complex entailments. Afterwards, we study the more expressive ΠLTL_f in which possibilistic statements can be recursively nested within other temporal expressions. We use the insights from ΠLTL_f^0 to construct an automata-based decision procedure for full ΠLTL_f.

These formalisms should provide the basis for a possibilistic extension of Declare and effectively managing more complex uncertainty statements in declarative process modelling.

2 Preliminaries

We start providing the background knowledge necessary for understanding our formalism; namely, the temporal logic LTL_f and the basics of possibility theory.

2.1 LTL_f

LTL_f [9] is a (discrete) linear temporal logic over finite traces. Syntactically, it looks exactly like the better known (infinite time) LTL [3, 28]. LTL_f *formulas* are constructed from a class $\mathcal{P}$ of propositional *variables* through the syntactic rule

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

where $x \in \mathcal{P}$. Briefly, $\bigcirc\varphi$ expresses that φ should hold in the *next* point in time, and $\varphi \,\mathcal{U}\, \psi$ expresses that φ holds from the current timepoint onwards *until* ψ is observed. The main difference with (classical) LTL is that time does not run infinitely, but eventually stops; that is, there is a final point in time, but no specific bound on when this point will be reached. All this is formalised through temporal models.

A *temporal model* is a finite sequence of propositional valuations, where as usual a propositional valuation is represented as a set of propositional variables. A valuation $\mathcal{V} \subseteq \mathcal{P}$ expresses which variables are *true* in it. Let $\mathcal{M} = \mathcal{V}_0\mathcal{V}_1 \cdots \mathcal{V}_n$ be a temporal model. The *satisfiability* relation of a formula φ w.r.t. $\mathcal{M}$ at time $k, 0 \leq k \leq n$ is defined inductively as follows:

- $\mathcal{M}, k \models x$ (where $x \in \mathcal{P}$) iff $x \in \mathcal{V}_k$;
- $\mathcal{M}, k \models \neg\varphi$ iff $\mathcal{M}, k \not\models \varphi$;
- $\mathcal{M}, k \models \varphi \wedge \psi$ iff $\mathcal{M}, k \models \varphi$ and $\mathcal{M}, k \models \psi$;
- $\mathcal{M}, k \models \bigcirc\varphi$ iff $k < n$ and $\mathcal{M}, k+1 \models \varphi$; and
- $\mathcal{M}, k \models \varphi \,\mathcal{U}\, \psi$ iff there exists $\ell, k \leq \ell \leq n$ such that $\mathcal{M}, \ell \models \psi$ and for all $m, k \leq m < \ell$, it holds that $\mathcal{M}, m \models \varphi$.

Intuitively, $\bigcirc\varphi$ is satisfied if φ is observed in the next timepoint, while the until operator is satisfied if there is a point in the future (up to the last timepoint) where ψ holds and in between φ is always satisfied. Note that this view on the until operator requires one to foresee the behaviour of the interpretation far into the future. To manage formulas of the form $\varphi \,\mathcal{U}\, \psi$ locally, one often considers the equivalent characterisation where $\mathcal{M}, k \models \varphi \,\mathcal{U}\, \psi$ iff (i) $\mathcal{M}, k \models \psi$ or (ii) $\mathcal{M}, k \models \varphi$ and $\mathcal{M}, k \models \bigcirc(\varphi \,\mathcal{U}\, \psi)$. Despite the apparent recursion on the last point, this semantics of $\varphi \,\mathcal{U}\, \psi$ is well defined and in particular $\mathcal{M}, n \models \varphi \,\mathcal{U}\, \psi$ holds whenever $\mathcal{M}, n \models \psi$.

The temporal model $\mathcal{M}$ *satisfies* the formula φ ($\mathcal{M} \models \varphi$) iff $\mathcal{M}, 0 \models \varphi$. A formula is *satisfiable* if there exists a temporal model that satisfies it. A set of formulas Γ is *consistent* iff there is a temporal model that satisfies all the formulas in Γ. From now on, we will often refer to temporal models simply as *models*, for brevity, and as *traces* in accordance to the terminology of process modelling [30].

LTL_f has been successfully used to represent processes, by introducing constraints on the traces which can be observed during the execution of the process [27]. Usually, these *process models* are constructed as a set of LTL_f formulas, although to improve readability and aid development, these formulas are often limited to a small class of general patterns, as is the case for the well known language Declare [27]. For complex, live, processes, constructing such a specification is far from obvious. One common approach, in this case, is to *mine* the trace logs for repeating patterns [29, 5]. Yet, due to the uncertainty of the real world and potential writing mistakes, it is not uncommon to produce inconsistent models in this manner. For instance, a constraint may allow for a limited number of exceptions (such as product defects) to happen randomly, but these exceptions will not be included in the mined process. Hence, there is an interest in developing extensions of LTL_f which can handle uncertainty.

2.2 Possibility Theory

Possibility theory [11] is an alternative to probability theory [4] developed for handling some kinds of uncertainty by measuring the *possibility* and *necessity* of events. Similarly to probability theory, each event is assigned a degree in $[0, 1]$, but possibility measures are typically easier to manipulate mainly because they represent a more *qualitative*, and less *quantitative*, notion of uncertainty. Indeed, the specific numbers assigned are less important than their relative ordering.

Formally, given a (possibly infinite) set Ω of *outcomes*, a *possibility measure* is a function $\pi : 2^\Omega \to [0, 1]$ which satisfies the following three properties:

1. $\pi(\emptyset) = 0$;
2. $\pi(\Omega) = 1$; and
3. if $U, V \subseteq \Omega$ are disjoint, then $\pi(U \cup V) = \max\{\pi(U), \pi(V)\}$.

Subsets of Ω (i.e., sets of outcomes) are called *events*. Equivalently, one can define a possibility measure as a function $\pi : \Omega \to [0, 1]$, which is later extended to events by setting $\pi(U) = \sup_{\omega \in U} \pi(\omega)$ for all $U \subseteq \Omega$. That is, possibility measures can be fully specified through a possibility value on the outcomes.

From this definition, one can immediately see the main difference between possibility and probability measures: for any set $U \subseteq \Omega$ a possibility measure π is such that $\max\{\pi(U), \pi(\Omega \setminus U)\} = 1$. This has two important implications. First, it could be the case that both, an event and its complement, have possibility 1, representing a situation where both situations are considered as equally possible. Second, whenever an event has a possibility strictly smaller than 1, its complement *must* have possibility 1. Intuitively, if one hesitates about the possibility of an event, it means that its opposite (i.e., its complement) is possible.

To emphasise the dissimilarity between the two kinds of measures of uncertainty, we also note that the constant function that maps every non-empty event to 1 (that is, $\pi(U) = 1$ for all $U \subseteq \Omega \setminus \{\emptyset\}$) is a possibility measure. This refers to a completely agnostic situation, where anything (except not observing any outcome) is possible. Moreover, if an event U is finite, one can verify that $\pi(U) \geq p$ for some $p \in [0, 1]$ simply by finding an outcome $\omega \in U$ such that $\pi(\omega) \geq p$.

Possibility measures implicitly define a dual *necessity measure* $N : 2^\Omega \to [0, 1]$ given by $N(U) = 1 - \pi(\Omega \setminus U)$ for all $U \subseteq \Omega$. The idea behind this measure is that if an event is more *necessary*, then its complement must be less possible; e.g., if it is necessary that it rains, then it must be impossible for it not to rain. We introduce this function here for completeness, but focus on possibility measures for the rest of the paper, for reasons that will become clear later.

In the next section we will combine the two notions of LTL_f and possibility theory to introduce a new temporal logic capable of dealing with (qualitative) uncertainty; but first, we briefly recall tree automata.

2.3 Tree Automata

A finite *tree* is a finite non-empty set of words of (positive) natural numbers $T \subseteq \mathbb{N}^*$ which is closed under prefixes and preceding siblings; that is, if $wi \in T$, then $w \in T$ and for all $1 \leq j \leq i$, $wj \in T$. Each finite tree T has a maximum number $k \in \mathbb{N}$ (called the *width*) such that $wk \in T$ for some $w \in \mathbb{N}^*$. The empty word ε is the *root*, and a *leaf* is a node $w \in T$ such that $w1 \notin T$. We will often consider *full* k-ary trees, where every non-leaf node has k successors.

A *labelling* of the tree T on a set Σ is a mapping $T \to \Sigma$. A tree which has been associated to a labelling is called a *labelled tree*. A *branch* of the tree T is a sequence $w_1, \ldots, w_n$ of nodes of T such that $w_1 = \varepsilon$, w_n is a leaf node, and for every $i, 1 \leq i < n$, $w_{i+1} = w_i m$ for some $m \in \mathbb{N}$. With a slight abuse of terminology, whenever a tree T is labelled, we call *branch* also the sequence of labels of a branch.

Definition 1. Let k be a positive natural number. A k-ary *tree automaton* is a tuple $\mathcal{A} = (Q, \Delta, I, F)$ where Q is a finite set of *states*, $I, F \subseteq Q$ are sets of *initial* and *final* states, respectively, and $\Delta \subseteq Q^{k+1}$ is the *transition relation*. A *run* of $\mathcal{A}$ on the (full) tree T of width k is a labelling $\rho : T \to Q$ such that $\rho(\varepsilon) \in I$ and for every $w \in T$, if w is not a leaf, then $(\rho(w), \rho(w1), \ldots, \rho(wk)) \in \Delta$. Such a run is called *successful* if for every leaf node $w \in T$, it holds that $\rho(w) \in F$. In that case, we say that $\mathcal{A}$ *accepts* the tree T. The *language accepted* by $\mathcal{A}$ is the set $\mathcal{L}(\mathcal{A})$ of finite trees for which there is a successful run of $\mathcal{A}$. The *emptiness problem* asks whether $\mathcal{L}(\mathcal{A}) = \emptyset$.

This definition focuses on a limited class of automata, known as *looping* automata, where transitions are not labelled but depend only on the current state [15]. This suffices for the purposes of this paper.

The emptiness problem of k-ary tree automata can be decided in time $\mathcal{O}(|Q|^{k+2})$ [31]. However, the automata constructed for reasoning within different logics often require an exponential number of states on the size of the input formulas (see Section 4), which yields an EXPTIME reasoning method. Under some conditions—in essence, polynomial arity, a polynomial description of each state, and a guarantee of polynomial depth on accepting runs—this bound can be improved to PSPACE [2]

3 ΠLTL$_f^0$

Taking inspiration from the simple probabilistic PLTL$_f^0$ [21, 23] (which is a special case of the more general PLTL$_f$), we define the possibilistic temporal logic on finite traces ΠLTL$_f^0$. Through this logic, one can introduce a set of constraints over the possibility of observing certain types of behaviours (called a knowledge base), which in turn defines a possibility measure over the class of all temporal models.

Definition 2. A ΠLTL$_f^0$ *knowledge base* (KB) is a finite set of constraints of the form $\langle \varphi \geq p \rangle$ or $\langle \varphi \leq p \rangle$ where φ is an LTL$_f$ formula and $p \in [0, 1]$. A *possibilistic model* is a pair $\mathfrak{I} = (\mathfrak{M}, \pi)$ where $\mathfrak{M}$ is a set of temporal models and π is a possibility measure over $\mathfrak{M}$. We say that the possibilistic model $\mathfrak{I}$ is *finite* iff $\mathfrak{M}$ is finite.

The *possibility* of an LTL$_f$ formula φ w.r.t. the possibilistic model $\mathfrak{I} = (\mathfrak{M}, \pi)$ is $\pi(\varphi) := \pi(\{\mathcal{M} \in \mathfrak{M} \mid \mathcal{M} \models \varphi\})$. $\mathfrak{I}$ *satisfies* the KB $\mathcal{K}$ ($\mathfrak{I} \models \mathcal{K}$) iff for every constraint $\langle \varphi \bowtie p \rangle \in \mathcal{K}$ (with $\bowtie \in \{\leq, \geq\}$), it holds that $\pi(\varphi) \bowtie p$. $\mathcal{K}$ is *consistent* iff there is at least one possibilistic model that satisfies it.

We note first that this definition slightly differs from standard approaches in possibilistic logics where formulas in a KB are typically interpreted as lower bounds for *necessity* degrees. By definition, we have that, if $N(\varphi) \geq p$ then $\pi(\neg\varphi) \leq 1 - p$. Hence, we decided to focus on interpreting KBs using possibility degrees only, but allowing both lower and upper bounds. This allows us also to include "weak" restrictions $\langle \varphi \geq p \rangle$ requiring at least one (but not necessarily all) temporal model in $\mathfrak{M}$ satisfying φ to have possibility degree at least p.

From now on, we will denote as $\Phi(\mathcal{K}) := \{\varphi \mid \langle \varphi \bowtie p \rangle \in \mathcal{K}\}$ be the set of all (classical) LTL$_f$ formulas appearing in $\mathcal{K}$ and as $\Pi(\mathcal{K}) := \{p \mid \langle \varphi \bowtie p \rangle \in \mathcal{K}\}$ the set of all possibility degrees appearing in $\mathcal{K}$.

According to Definition 2, a model $\mathfrak{I}$ may contain infinitely many traces (with an adequate possibility measure over them). Yet, if one is interested in KB consistency, it is only important how these traces behave over the formulas that appear in the KB. This yields our first result: ΠLTL$_f^0$ has the *finite model property*.

Theorem 3. *Let $\mathcal{K}$ be a ΠLTL$_f^0$ KB. $\mathcal{K}$ is consistent iff there is a* finite *possibilistic model that satisfies $\mathcal{K}$.*

Proof. For the "if" direction, the result is trivial: if there is a finite possibilistic model satisfying $\mathcal{K}$, then $\mathcal{K}$ is consistent by definition.

Conversely, if $\mathcal{K}$ is consistent, then there is a possibilistic model $\mathfrak{I} = (\mathfrak{M}, \pi)$ that satisfies $\mathcal{K}$. We construct an equivalence relation $\sim$ over $\mathfrak{M}$ as follows: for $\mathcal{M}, \mathcal{M}' \in \mathfrak{M}$ we have that $\mathcal{M} \sim \mathcal{M}'$ iff for every $\varphi \in \Phi(\mathcal{K})$ it holds that $\mathcal{M} \models \varphi$ iff $\mathcal{M}' \models \varphi$; i.e., two temporal models are equivalent if they satisfy the same class of formulas in $\Phi(\mathcal{K})$. This equivalence relation induces a new possibilistic model $\mathfrak{I}_\sim = (\mathfrak{M}/\sim, \pi_\sim)$ where $\mathfrak{M}/\sim$ is the quotient set of $\mathfrak{M}$ w.r.t. $\sim$,[1] and for every equivalence class $[\mathcal{M}]_\sim$ we define $\pi_\sim([\mathcal{M}]_\sim) := \sup_{\mathcal{M}' \sim \mathcal{M}} \pi(\mathcal{M}')$. Note that $\sim$ has at most 2^n equivalence classes, where n is the cardinality of $\Phi(\mathcal{K})$; hence $\mathfrak{I}_\sim$ is a finite model. It remains to be shown that $\mathfrak{I}_\sim \models \mathcal{K}$. Consider a constraint $\langle \varphi \bowtie p \rangle \in \mathcal{K}$. By construction,[2]

$$\begin{aligned}
\pi_\sim(\varphi) &= \pi_\sim(\{[\mathcal{M}]_\sim \in \mathfrak{M}/\sim \mid \mathcal{M} \models \varphi) \\
&= \sup_{\mathcal{M}' \sim \mathcal{M}, \mathcal{M} \models \varphi} \pi(\mathcal{M}') = \sup_{\mathcal{M} \models \varphi} \pi(\mathcal{M}') \\
&= \pi(\{\mathcal{M} \in \mathfrak{M} \mid \mathcal{M} \models \varphi\}) \\
&= \pi(\varphi).
\end{aligned}$$

This means that $\mathfrak{I} \models \langle \varphi \bowtie p \rangle$ iff $\mathfrak{I}_\sim \models \langle \varphi \bowtie p \rangle$. Since $\mathfrak{I}$ is a model of $\mathcal{K}$, so is $\mathfrak{I}_\sim$. □

The construction used to prove this theorem suggests also a way to effectively decide consistency: one can try to construct a temporal interpretation satisfying each of the 2^n combinations of formulas in $\mathcal{K}$ and assign an adequate possibility degree according to the constraints whenever this sub-KB is (classically) consistent. Thus, we can conclude that ΠLTL$_f^0$ KB consistency is decidable in exponential time: it requires at most exponentially many LTL$_f$ consistency tests, each of which runs in PSPACE [9]. However, this approach requires us to verify many unnecessary cases.

As a first step, note that by definition, all constraints of the form $\langle \varphi \geq p \rangle$ can be treated independently, since they can be satisfied with different temporal models associated to a high possibility degree. Consider for instance the KB $\mathcal{K} = \{\langle x \geq 0.9 \rangle, \langle \neg x \geq 0.9 \rangle\}$, which is consistent although there is no trace that satisfies both formulas. A possibilistic model satisfying this KB will have (at least) two temporal models $\mathcal{M}_1, \mathcal{M}_2$ such that $\mathcal{M}_1 \models x$, $\mathcal{M}_2 \models \neg x$, and both models have possibility degree at least 0.9 (in fact, one of them will have possibility 1). Thus, lower possibility bounds may only be problematic when combined with upper bounds.

On the other hand, upper bounds have a very different behaviour. For example, the KB $\mathcal{K} = \{\langle x \leq 0.9 \rangle, \langle \neg x \leq 0.9 \rangle\}$, which looks

[1] That is, the set of all equivalence classes of $\mathfrak{M}$ w.r.t. $\sim$.

[2] The choice of the representative for a class is irrelevant, as all models in $[\mathcal{M}]_\sim$ are indistinguishable from the point of view of $\Phi(\mathcal{K})$.

very similar to the previous example, is inconsistent, because it constraints all traces to have a possibility degree strictly smaller than 1, which is impossible–recall the definition of a possibility measure. This means that, contrary to lower bounds, upper possibility bounds alone can lead to inconsistencies. To make this more clear, we emphasise once again that the axiom $\langle \varphi \leq p \rangle$ also implicitly requires the existence of a temporal model satisfying $\neg\varphi$ with possibility 1. Yet, note that the conflict observed may require more than two formulas to appear. These insights will allow us to reduce the complexity of KB consistency from the previously mentioned exponential time to "only" polynomial space.

We partition the KB $\mathcal{K}$ into two classes: $\mathcal{K}^+$ which contains all the constraints of the form $\langle \varphi \geq p \rangle$ with $p > 0$ (from now on called *positive constraints*) and $\mathcal{K}^-$ containing the constraints of the form $\langle \varphi \leq p \rangle$ with $p < 1$ (called *negative constraints*). Note that positive constraints with $p = 0$ and negative constraints with $p = 1$ are uninformative; for that reason we do not consider them in the following analysis. As previously explained, a positive constraint requires the *existence* of a trace (satisfying a formula) with a possibility larger or equal to some value; dually, negative constraints express that *all* traces satisfying the associated formula *must* have a possibility degree under a given threshold.

Note first that if any positive constraint with $p > 0$ is given by a contradictory formula, then $\mathcal{K}$ is inconsistent. For example, the KB with the constraint $\langle x \wedge \neg x \geq 0.5 \rangle$ is inconsistent, because its satisfaction requires the existence of a trace which satisfies the contradiction $x \wedge \neg x$, which is impossible. Similarly, if any negative constraint with $p < 1$ uses a tautology, then $\mathcal{K}$ is inconsistent. For example, the constraint $\langle x \vee \neg x \leq 0.5 \rangle$ cannot be satisfied by any possibilistic model because every trace satisfies $x \vee \neg x$ and the constraint requires that all such traces have possibility at most 0.5, which is not allowed in a possibility measure. These situations can be easily verified during the construction of the KB, and hence disregard them as *trivial* cases as formalised next.

Definition 4. The KB $\mathcal{K}$ is called *trivial* iff $\Phi(\mathcal{K}^+)$ contains a contradiction or $\Phi(\mathcal{K}^-)$ contains a tautology.

For non-trivial KBs, it suffices to consider some simple combinations of formulas to decide consistency, as characterised in the next theorem. But first, we introduce some useful notation. Given a set $\mathcal{L}$ of negative constraints, we denote as $\mathcal{L}_p := \{\langle \varphi \leq q \rangle \in \mathcal{L} \mid q < p\}$ the subset of $\mathcal{L}$ containing only constraints with a possibility degree smaller than p and as $\psi_p^{\mathcal{L}} := \bigvee_{\varphi \in \Phi(\mathcal{L}_p)} \varphi$ the disjunction of all formulas appearing in this set. For brevity, $\psi^{\mathcal{L}}$ will denote $\psi_1^{\mathcal{L}}$; that is, the disjunction of all formulas in $\mathcal{L}$. If $\mathcal{L}$ is a set of positive constraints, then $\mathcal{L}_p := \{\langle \varphi \geq q \rangle \in \mathcal{L} \mid p < q\}$ is the subset of $\mathcal{L}$ containing only constraints with a possibility degree greater than p.

Theorem 5. *Let $\mathcal{K}$ be a non-trivial ΠLTL$_f^0$ KB. $\mathcal{K}$ is inconsistent iff*

1. *$\psi^{\mathcal{K}^-}$ is a tautology; or*
2. *there exist $\langle \varphi \geq p \rangle \in \mathcal{K}^+$ such that $\varphi \wedge \neg\psi_p^{\mathcal{K}^-}$ is a contradiction.*

Proof. [$\Leftarrow$] Consider first the second point. If $\langle \varphi \geq p \rangle \in \mathcal{K}^+$ then any model satisfying the KB $\mathcal{K} = \mathcal{K}^+ \cup \mathcal{K}^-$ must have a temporal model $\mathcal{M}$ such that $\mathcal{M} \models \varphi$ and $\pi(\mathcal{M}) \geq p$. For this model to also satisfy $\mathcal{K}_p^-$ it must hold that, for every $\langle \varphi_2 \leq p_2 \rangle \in \mathcal{K}_p^-$, $\mathcal{M} \not\models \varphi_2$ (recall that if $\mathcal{M} \models \varphi_2$ then $\pi(\varphi_2) \geq \pi(\mathcal{M}) > p_2$). This means that $\mathcal{M} \models \bigwedge_{\psi \in \mathcal{K}_p^-} \neg\psi = \neg\psi_p^{\mathcal{K}^-}$. But that is impossible because the formula $\varphi \wedge \neg\psi_p^{\mathcal{K}^-}$ is a contradiction. Hence, $\mathcal{K}$ is inconsistent.

For the first point, if a possibilistic model $\mathfrak{I} = (\mathfrak{M}, \pi)$ satisfies $\mathcal{K}^-$, then every temporal model $\mathcal{M} \in \mathfrak{M}$ such that $\mathcal{M} \models \psi^{\mathcal{K}^-}$ must be such that $\pi(\mathcal{M}) \leq \min_{p \in \Pi(\mathcal{K}^-)} p < 1$. Since π is a possibility measure, there must exist some $\mathcal{M}_0 \in \mathfrak{M}$ with $\pi(\mathcal{M}_0) = 1$ and, by the previous argument $\mathcal{M}_0 \not\models \psi^{\mathcal{K}^-}$. But that is impossible because $\psi^{\mathcal{K}^-}$ is a tautology by assumption. Hence $\mathcal{K}$ is inconsistent.
[$\Rightarrow$] For the converse, assume that none of the two conditions hold; we will show that $\mathcal{K}$ must be consistent by building a satisfying possibilistic model. If $\mathcal{K}^- = \emptyset$, then for each $\langle \varphi \geq p \rangle \in \mathcal{K}^+$ construct a temporal model satisfying φ and assign to all these temporal models possibility degree 1. This clearly defines a possibilistic model satisfying $\mathcal{K}$. If $\mathcal{K}^- \neq \emptyset$, then for each $\langle \varphi \geq p \rangle \in \mathcal{K}^+$ we construct a temporal model which satisfies $\varphi \wedge \neg\psi_p^{\mathcal{K}^-}$ (which must exist by assumption) and assign it possibility p. We also construct a temporal model satisfying $\neg\psi^{\mathcal{K}^-}$ (which again, exists because the first condition does not hold) and assign it possibility 1. To show that this is indeed a model of $\mathcal{K}$ we only need to verify that it does not violate any negative constraint (all positive constraints are satisfied by construction). Suppose that there is some $\langle \psi \leq q \rangle \in \mathcal{K}^-$ and some temporal model $\mathcal{M}$ such that $\mathcal{M} \models \psi$ and $\pi(\mathcal{M}) > q$. By construction, $\mathcal{M}$ was generated by a positive constraint $\langle \varphi \geq p \rangle \in \mathcal{K}^+$ with $p > q$. But then $\langle \psi \leq q \rangle \in \mathcal{K}_p^-$ but then $\mathcal{M} \not\models \psi$ because $\mathcal{M} \models \neg\psi_p^{\mathcal{K}^-}$. Thus, no negative constraint can be violated. □

The consequence of this theorem is that ΠLTL$_f^0$ KB consistency can be decided quite efficiently, simply by making a linear number of calls to a standard LTL$_f$ reasoner. Indeed, if $\mathcal{K}$ has at m positive constraints and n negative constraints, then one needs to check satisfiability of $m + n$ formulas (to guarantee non-triviality) and at most $m + 1$ formulas to verify the conditions of Theorem 5. Since each such check can be made using only polynomial space, we overall obtain that ΠLTL$_f^0$ KB consistency is PSPACE-complete (just as in classical LTL$_f$).

Interestingly, this idea is not only parallelisable but, if the LTL$_f$ satisfiability algorithm is modular (as in the case of automata-based procedures; see Section 4) the consistency method can be implemented quite efficiently, as it is based on mixing and matching formulas which are fixed at the input. Moreover, as can be seen from Theorem 5 and its proof, one can often reduce the number of tests further.

Corollary 6. *Let $\mathcal{K}$ be a nontrivial KB and $n_{\mathcal{K}} := \min_{p \in \Pi(\mathcal{K}^-)} p$. $\mathcal{K}$ is consistent iff $\mathcal{K}^- \cup \mathcal{K}^+_{n_{\mathcal{K}}}$ is consistent.*

Overall, this means that one needs to do at most $|\mathcal{K}^+_{n_{\mathcal{K}}}| + 1$ satisfiability checks to decide KB consistency and, in particular, if $\max_{p \in \Pi(\mathcal{K}^+)} p \leq \min_{p \in \Pi(\mathcal{K}^-)} p$ it is only necessary to check that $\psi^{\mathcal{K}^-}$ is not a tautology.

3.1 Inconsistency Degrees

As usual in possibilistic logics, when a KB is inconsistent, we can provide a measure of the uncertainty level at which inconsistencies happen. Since we have both, positive and negative constraints which interact with each other, we compute a class of pairs of inconsistency degrees, rather than a single value, as defined next.

Definition 7. Given a KB $\mathcal{K}$ and $p, q \in [0, 1]$, define the sub-KB $\mathcal{K}_{p,q} := \mathcal{K} \setminus (\mathcal{K}_p^+ \cup \mathcal{K}_q^-)$ which contains all positive constraints with degree up to p and all negative constraints with degree at least q.

Algorithm 1: Enumeration of optimal inconsistency degrees

Data: $\mathcal{K}$ inconsistent ΠLTL_f^0 KB
Result: Optimal inconsistency degrees of $\mathcal{K}$

1 $p_0 \leftarrow 0$; $p_{\max} \leftarrow \max\{p \in \Pi(\mathcal{K}^+)\}$
2 $q_0 \leftarrow 0$
3 **for** $q \in \Pi(\mathcal{K}^-)$ **do**
4 **if** $\mathcal{K}_{p_0,q}$ *is consistent* **then**
5 **return** (p_0, q_0)
6 $q_0 \leftarrow q$
7 **while** $\mathcal{K}_{p_0,q_0}$ *is consistent and* $p_0 < p_{\max}$ **do**
8 $p_0 \leftarrow next(\Pi(\mathcal{K}^+))$
9 **end**
10 **end**
11 **end**

The pair (p, q) is called an *inconsistency degree* for $\mathcal{K}$ iff $\mathcal{K}_{p,q}$ is inconsistent. It is an *optimal inconsistency degree* if for all p', q' such that $p' < p$ and $q < q'$, $\mathcal{K}_{p',q}$ and $\mathcal{K}_{p,q'}$ are consistent.

Note that there can be many optimal inconsistency degrees. In fact, the definition uses a Pareto condition which commonly yields a full class of optimal elements.

Example 8. Consider the inconsistent KB

$$\mathcal{K} = \{\langle x \geq 0.4\rangle, \langle \bigcirc y \geq 0.9\rangle, \langle \neg \bigcirc y \leq 0.7\rangle, \langle \neg \bigcirc \neg y \leq 0.8\rangle\}.$$

Clearly, $\mathcal{K}_{0.9,0.7} = \mathcal{K}$ is inconsistent, and hence $(0.9, 0.7)$ is an inconsistency degree. However, it is not optimal. Indeed, $\mathcal{K}_{0.9,0.8}$ and $\mathcal{K}_{0,0.7}$ are also inconsistent. It is easy to verify using Corollary 6 that $(0.9, 0.8)$ and $(0, 0.7)$ are the optimal inconsistency degrees.

One interesting insight which arises from the definition (and is visible in the example) is that if (p, q) is an optimal consistency degree, then $p \in \Pi(\mathcal{K}^+) \cup \{0\}$ and $q \in \Pi(\mathcal{K}^-)$. Moreover, by Corollary 6, if $p \neq 0$, then $p > q$. Using this information, we can conclude that (i) there are at most $|\Pi(\mathcal{K}^-)|$ different optimal inconsistency degrees, and (ii) they can be efficiently enumerated as described in Algorithm 1.

The algorithm starts with an empty set of positive constraints and all negative constraints. While the subKB remains inconsistent, it tries to further remove negative constraints until an optimal inconsistency degree is found. The next possible optimal inconsistency degree will need to include some positive constraints; the exact value at which to continue the search for optimality is found in the while loop (line 7). The correctness of this algorithm is guaranteed by the previously described insights. As with deciding consistency, enumerating all optimal inconsistency degrees consumes only polynomial space (although it may require an exponential execution time).

The importance of optimal inconsistency degrees is that they compactly express the uncertainty required to observe an inconsistency in the KB. Note that for positive constraints $\langle \varphi \geq p\rangle$, the larger the p the more restrictive the constraint is: it requires a higher degree for some temporal models. Conversely, negative constraints $\langle \varphi \leq p\rangle$ are more restrictive when p is smaller: all temporal models of a certain kind must have a lower possibility degree. When dealing with uncertainty, a more restrictive clause refers to a higher certainty on its correctness. Thus, optimal inconsistency degrees refer to the minimal certainty a user needs to require to observe the inconsistency; remaining at a "lower" level of uncertainty recovers consistency of the knowledge.

3.2 Entailments

As usual in logical languages, and in particular in the context of knowledge representation and reasoning, consistency is but one of the many reasoning problems that can be considered. Another important problem, which is helpful also for managing process models, is that of *entailment*. We say that the KB $\mathcal{K}$ *entails* the constraint $\langle \varphi \bowtie p\rangle$ (denoted by $\mathcal{K} \models \langle \varphi \bowtie p\rangle$) iff every possibilistic model that satisfies $\mathcal{K}$ is such that $\pi(\varphi) \bowtie p$. This problem can be reduced to consistency using standard techniques, and hence is decidable in PSPACE as well.

Theorem 9. *Let $\mathcal{K}$ be a ΠLTL_f^0 KB, φ an LTL_f formula, and $p \in [0, 1]$. (i) $\mathcal{K} \models \langle \varphi \geq p\rangle$ iff $\mathcal{K} \cup \{\langle \varphi < p\rangle\}$ is inconsistent and (ii) $\mathcal{K} \models \langle \varphi \leq p\rangle$ iff $\mathcal{K} \cup \{\langle \varphi > p\rangle\}$ is inconsistent.*

Note that in this theorem we use strict lower and upper bounds—$\langle \varphi < p\rangle$ and $\langle \varphi > p\rangle$—for the constraints introduced to KBs, but these are not allowed in the language of ΠLTL_f^0 as given in Definition 2. However, given the results on consistency checking, and in particular Corollary 6, we can actually rewrite the condition of Theorem 9 without strict inequalities. The proof of the following proposition is straightforward.

Proposition 10. *Let $\mathcal{K}$ be a KB, φ an LTL_f formula, and $p \in [0, 1]$. Let q be the smallest element in $\Pi(\mathcal{K}^-)$ such that $p < q$ (or 1 if there is none). Then $\mathcal{K} \cup \{\langle \varphi < p\rangle\}$ is consistent iff $\mathcal{K} \cup \{\langle \varphi \leq q\rangle\}$ is consistent. Analogously, if q is the biggest element in $\Pi(\mathcal{K}^+)$ such that $q < p$ (or 0 if there is none), then $\mathcal{K} \cup \{\langle \varphi > p\rangle\}$ is consistent iff $\mathcal{K} \cup \{\langle \varphi \geq q\rangle\}$.*

This concludes our analysis of the simple ΠLTL_f^0 based on a possibility distribution over temporal models, which is only measured from the beginning of the execution. In the next section, we will use the insights gained from this language to study a more expressive logic, which allows for a possibilistic constructor which can be nested with other expressions.

4 Full ΠLTL_f

We now extend ΠLTL_f^0 to a possibilistic variant of the probabilistic temporal logic PLTL_f. As mentioned already, in this logic we allow for a constructor constraining the possibility degrees of the future observations of the execution.

Syntactically, ΠLTL_f formulas are built through the grammar rule

$$\varphi ::= x \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \circledast_{\bowtie p}\varphi,$$

where $x \in \mathcal{P}$, $p \in [0, 1]$, and $\bowtie \in \{\leq, \geq, <, >\}$. A ΠLTL_f KB is a finite set of ΠLTL_f formulas.

Thus, the syntax of ΠLTL_f extends LTL_f with expressions of the form $\circledast_{\bowtie p}\varphi$ which intuitively express that *in the next point in time* the possibility of φ is $\bowtie p$. Importantly, recall that if $\bowtie \in \{\leq, <\}$ this also provides some knowledge about the possibility of $\neg\varphi$. The formal semantics of this logic is based on finite trees, whose nodes are associated with propositional valuations and a possibility measure through two labelling functions.

Definition 11. A ΠLTL_f *interpretation* is a triple $I = (T, \cdot^I, P)$, where T is a finite tree, $\cdot^I : T \to \mathcal{P}(\mathcal{P})$ is a labelling of T on the set of propositional valuations,[3] and $P : T \setminus \{\varepsilon\} \to [0, 1]$ is such that

[3] As usual, we describe a propositional valuation by the set of variables it makes true.

for each non-leaf node $w \in T$, $\max_{wi \in T} P(wi) = 1$. *Satisfiability* of a formula in a tree node is defined inductively, extending the LTL_f semantics. For an interpretation $I = (T, \cdot^I, P)$ and $w \in T$:

- $I, w \models x$ iff $x \in w^I$;
- $I, w \models \neg\varphi$ iff $I, w \not\models \varphi$;
- $I, w \models \varphi \wedge \psi$ iff $I, w \models \varphi$ and $I, w \models \psi$;
- $I, w \models \bigcirc\varphi$ iff w is not a leaf node and for all $i \in \mathbb{N}$, if $wi \in T$ then $I, wi \models \varphi$;
- $I, w \models \varphi \,\mathcal{U}\, \psi$ iff one of the following holds: (i) $I, w \models \psi$ or (ii) $I, w \models \varphi$ and $I, w \models \bigcirc(\varphi \,\mathcal{U}\, \psi)$; and
- $I, w \models \circledast_{\bowtie p}\varphi$ iff $\max_{wi \in T; I, wi \models \varphi} P(wi) \bowtie p$.[4]

I is a *model* of ϕ ($I \models \varphi$) if $I, \varepsilon \models \varphi$; it is a *model* of the KB $\mathcal{K}$ (denoted $I \models \mathcal{K}$) iff $I \models \varphi$ for all $\varphi \in \mathcal{K}$. $\mathcal{K}$ is *consistent* if it has a model.

As it can be seen, ΠLTL_f^0 is a special case of ΠLTL_f where the constructor $\circledast$ is only applied to LTL_f formulas and strict inequalities $<$ and $>$ are never used. The idea of the function P is that it assigns the possibility measure for the remaining of the execution of the traces (and hence there must be one of them that is assigned possibility 1). Note that these degrees refer to the *future* of the execution, regardless of what has happened before in time. This is consistent with the usual assumptions from LTL_f.

The insights obtained in the previous section will be helpful for building an automaton which can be used to decide consistency of ΠLTL_f KBs. This automaton adapts the ideas developed for LTL_f [9], but most handle the tree structures and the possibility measure of the semantics of ΠLTL_f.

Given a ΠLTL_f KB $\mathcal{K}$, let $\mathsf{csub}(\mathcal{K})$ be the smallest set of formulas such that: $\mathcal{K} \subseteq \mathsf{csub}(\mathcal{K})$; is closed under subformulas and negation (modulo removal of double negations); and if $\varphi \,\mathcal{U}\, \psi \in \mathsf{csub}(\mathcal{K})$, then $\bigcirc(\varphi \,\mathcal{U}\, \psi) \in \mathsf{csub}(\mathcal{K})$. We rewrite all possibilistic formulas appearing in $\mathsf{csub}(\mathcal{K})$ to have the form $\circledast_{\geq p}\varphi$ or $\circledast_{\leq p}\varphi$. This can be done without loss of generality because $\neg\circledast_{\geq p}\varphi \equiv \circledast_{<p}\varphi$ and, as seen in the previous section, once $\mathcal{K}$ is fixed, the latter formula can be equivalently written to the form $\circledast_{\leq p'}\varphi$ for some $p' < p$ (and analogously for negations of upper bounds).

As we will see, the formulas in $\mathsf{csub}(\mathcal{K})$ are the only ones that need to be taken into account when trying to construct a model of $\mathcal{K}$. We need to identify which sets of these formulas can be satisfied together in a node, forming what are known as *types*.

Definition 12. A subset $\tau \subseteq \mathsf{csub}(\mathcal{K})$ is a *type* iff (i) for every $\neg\varphi \in \mathsf{csub}(\mathcal{K})$, $\varphi \in \tau$ iff $\neg\varphi \notin \tau$; (ii) for every $\varphi \wedge \psi \in \mathsf{csub}(\mathcal{K})$, $\varphi \wedge \psi \in \tau$ iff $\{\varphi, \psi\} \subseteq \tau$; and (iii) for every $\varphi \,\mathcal{U}\, \psi \in \mathsf{csub}(\mathcal{K})$, $\varphi \,\mathcal{U}\, \psi \in \tau$ iff $\psi \in \tau$ or $\{\varphi, \bigcirc(\varphi \,\mathcal{U}\, \psi)\} \subseteq \tau$.

We use $\mathsf{type}(\mathcal{K})$ to denote the set of all types of $\mathcal{K}$.

In words, a type is a maximally consistent subset of $\mathsf{csub}(\mathcal{K})$ which in addition is consistent with the temporal interpretation of the *until* operator. This yields a local view on a possible model of $\mathcal{K}$, but still does not show how to deal with $\bigcirc$ or $\circledast$. For these we will need to find a way to deal with the branching of the trees. Recall from Section 3 that finding a satisfying model in the presence of n positive constraints requires constructing at most $n + 1$ different temporal models. We use the same idea, but adapted to the branches in the tree-shaped model.

[4] If w has no successor node that satisfies φ—in particular if w is a leaf node—then $\max_{wi \in T; I, wi \models \varphi} P(wi) = 0$ as standard in recursive operations.

Consider the sets $P(\mathcal{K}) := \{\circledast_{\geq p}\varphi \in \mathsf{csub}(\mathcal{K})\}$ of *positive constraints* and $N(\mathcal{K}) := \{\circledast_{\leq p}\varphi \in \mathsf{csub}(\mathcal{K})\}$ of *negative constraints*, and let $n_\mathcal{K} := |P(\mathcal{K})|$. We assume that the elements of $P(\mathcal{K})$ are enumerated in some arbitrary (but fixed) manner so that $P(\mathcal{K}) = \{\psi_1, \ldots, \psi_{n_\mathcal{K}}\}$; abusing the notation, we will also call $p_1, \ldots, p_{n_\mathcal{K}}$ the possibility degrees of these positive constraints and $\chi_i, 1 \leq i \leq n_k$ their associated formulas. That is, each ψ_i is of the form $\circledast_{\geq p_i}\chi_i$. For a given value $p \in [0, 1]$, and a set $\sigma \subseteq \mathsf{csub}(\mathcal{K})$, we define $\Xi(\sigma, p) := \{\neg\varphi \mid \circledast_{\leq q}\varphi \in \sigma, q < p\}$ the set of the negated formulas associated to negative contraints to a degree below p. We set $\Xi(\sigma) := \Xi(\sigma, 1)$ for brevity.

With this notation, we can now define compatible tuples of types, which will serve as the transition relation in the automaton.

Definition 13. Let $\tau, \tau_1, \ldots, \tau_{n_\mathcal{K}+1} \in \mathsf{type}(\mathcal{K})$. We say that the tuple $(\tau_1, \ldots, \tau_{n_\mathcal{K}+1})$ is *compatible* with τ iff

1. for every $\bigcirc\varphi \in \mathsf{csub}(\mathcal{K})$, $\bigcirc\varphi \in \tau$ iff for every $i, 1 \leq i \leq n_\mathcal{K}+1$ $\varphi \in \tau_i$
2. for every i, $1 \leq i \leq n_\mathcal{K}$, $\psi_i \in \tau$ iff $\{\chi_i\} \cup \Xi(\tau, p_i) \subseteq \tau_i$; and
3. $\Xi(\tau) \subseteq \tau_{n_\mathcal{K}+1}$

The idea is that the type τ_i is used to satisfy the positive constraint ψ_i while not violating any relevant negative constraint (hence the requirement that $\Xi(\tau, p_i) \subseteq \tau_i$. The information of $\bigcirc$ is propagated to all successors in the tree. Finally, one last successor makes sure that a possibility distribution is built (adding a new branch which will intuitively have possibility 1) without violating any of the negative constraints. With all these notions, we can now construct an automaton for deciding KB consistency.

Given a KB $\mathcal{K}$, we define the automaton $\mathcal{A}_\mathcal{K} = (Q, \Delta, I, F)$ of arity $n_\mathcal{K} + 1$ where $Q = \mathsf{type}(\mathcal{K})$; $\Delta \subseteq Q^{n_\mathcal{K}+2}$ contains all tuples $(\tau, \tau_1, \ldots, \tau_{n_\mathcal{K}})$ such that $(\tau_1, \ldots, \tau_{n_\mathcal{K}})$ is compatible with the type τ; $I = \{\tau \in \mathsf{type}(\mathcal{K}) \mid \mathcal{K} \subseteq \tau\}$; and F is the set of all types that do not contain any formula of the form $\bigcirc\varphi \in \mathsf{csub}(\mathcal{K})$ or $\circledast_{\geq p}\varphi$. We thus get the main result of this section.

Theorem 14. *The ΠLTL_f KB $\mathcal{K}$ is consistent iff the automaton $\mathcal{A}_\mathcal{K}$ is not empty.*

Proof (sketch). [$\Leftarrow$] Let $\rho : T \to Q$ be a successful run of $\mathcal{A}_\mathcal{K}$. We construct an interpretation $I = (T, \cdot^I, P)$ over the same tree T as follows: (i) for each node $w \in T$, $w^I = \rho(w) \cap \mathcal{P}$; (ii) for each node $wi \in T$ with $i \leq n_\mathcal{K}$, if $\psi_i \in \rho(w)$, then $P(wi) = p_i$; otherwise, $P(wi) = 0$; and (iii) for each node $w(n_\mathcal{K} + 1) \in T$, $P(w(n_\mathcal{K} + 1)) = 1$. It can be shown by a double induction (on the structure of the formula and the structure of the tree) that for every node $w \in T$ and every formula $\varphi \in \mathsf{csub}(\mathcal{K})$, $I, w \models \varphi$ iff $\varphi \in \rho(w)$. Since ρ is a successful run, we know that $\mathcal{K} \subseteq \rho(\varepsilon)$ and hence $I \models \varphi$ for all $\varphi \in \mathcal{K}$, showing that $\mathcal{K}$ is consistent.

[$\Rightarrow$] Conversely, let $I = (T, \cdot^I, P)$ be a model of $\mathcal{K}$. We recursively construct a successful run $\rho : T' \to Q$ of $\mathcal{A}_\mathcal{K}$ as follows. First we set $\rho(\varepsilon) := \{\varphi \in \mathsf{csub}(\mathcal{K}) \mid I \models \varphi\}$ and initialise a function $f : T' \to T$ as $f(\varepsilon) = \varepsilon$. This function keeps the invariant $\varphi \in \rho(w)$ iff $I, f(w) \models \varphi$. For the recursion, given a node w for which ρ and f are already defined, if $f(w)$ is not a leaf node, then for every i, $1 \leq i \leq n_\mathcal{K}$ such that $\psi_i \in \rho(w)$, find a successor v of $f(w)$ in T such that $I, v \models \chi_i$ and $P(v) \geq p_i$ (which must exist since I is a model) and set $\rho(wi) := \{\varphi \in \mathsf{csub}(\mathcal{K}) \mid I, v \models \varphi\}$ and $f(wi) := v$. For all other indices $j, 1 \leq j \leq n_\mathcal{K} + 1$, find a successor u of $f(w)$ in T such that $P(u) = 1$ (which must also exist since I is a model) and set $\rho(wj) := \{\varphi \in \mathsf{csub}(\mathcal{K}) \mid I, u \models \varphi\}$ and $f(wj) := u$. It can be checked that every node of the tree T'

constructed in this way is labelled with a type, and that all transitions are compatible. Moreover, $\mathcal{K} \subseteq \rho(\varepsilon)$ because I satisfies all formulas in $\mathcal{K}$ and leaf nodes cannot contain any formula of the form $\bigcirc\varphi$ because then I would not satisfy it. Hence ρ is a successful run of $\mathcal{A}_\mathcal{K}$ and the automaton is not empty. □

This theorem shows that KB consistency is reducible to an emptiness test of tree automata. As mentioned in the preliminaries, automata emptiness can be decided in time $\mathcal{O}(|Q|^{k+2})$. In our case, Q is the set of all types, and its cardinality is bounded exponentially on the number of formulas in $\mathcal{K}$. Perhaps more problematic is that the arity of the automaton also depends on the input KB (it is given by the number of positive constraints in $\mathsf{csub}(\mathcal{K})$). However, this does not mean that we end up with a double exponential procedure. In fact, the linear exponent based on $n_\mathcal{K}$ does not affect the overall complexity.

Corollary 15. *ΠLTL$_f$ KB consistency is in* EXPTIME.

On the other hand, although the arity of the automaton $n_\mathcal{K} + 1$ and the representation of each state are polynomial on the size of $\mathcal{K}$, we cannot bound the depth of successful runs polynomially on this same measure. Note that this is already true for classical LTL$_f$, where formulas may be only satisfiable in temporal models of exponential length. Thus, $\mathcal{A}_\mathcal{K}$ is not a candidate for a PSPACE on-the-fly construction as proposed in [2]. To the best of our efforts, we have been unable to show that ΠLTL$_f$ consistency is EXPTIME-hard, but given the characteristics of ΠLTL$_f$ models, it would be surprising if the complexity was lower.

Before concluding, we provide some remarks about the expressivity of ΠLTL$_f$. First note that LTL$_f$ is a special case of both ΠLTL$_f^0$ and ΠLTL$_f$ not only in syntactic but also in semantic terms. Specifically, an LTL$_f$ formula is also a ΠLTL$_f$ formula where the constructor $\circledast$ is not used, and can be seen as the ΠLTL$_f^0$ KB $\{\langle\varphi \geq 1\rangle\}$. By Theorems 5 and 14, such a formula has a satisfying temporal model iff it is consistent as a ΠLTL$_f^0$ KB iff it has a ΠLTL$_f$ model with branching one (that is, a degenerate tree where nodes have at most one successor)—which is exactly the semantics of LTL$_f$.

Given the tree-shaped semantics of ΠLTL$_f$, one could be tempted to believe that there is a connection with CTL [8] or its variant on finite time. However, the previous observation shows that ΠLTL$_f$ cannot be contained in CTL: there exist LTL$_f$ formulas (expressible in ΠLTL$_f$), which cannot be expressed in CTL [7, 13]. Conversely, since the only way to guarantee that something holds in *all* successors is through the "next" operator ($\bigcirc$), not all CTL formulas are expressible in ΠLTL$_f$ either.

5 Conclusions

In this paper, we have studied possibilistic extensions of the linear temporal logic over finite time LTL$_f$ as a way to effectively manage qualitative uncertainty in temporal KBs. Although the work focuses on the logical backbone of the formalism and its reasoning problems and methods, our motivation is not only theoretical. We are also motivated by applications in declarative process modelling.

Indeed, LTL$_f$ is one of the main languages used for process modelling, with the well-known Declare [27] being a simplified syntactic variant of it. Specifically, a Declare specification—given as a set of Declare constraints—is, in its back-end, nothing more than a set of LTL$_f$ formulas. Recent attention has been given to the question of dealing with uncertainty in specifications over real domains. One hardly needs to argue that real-world processes are surrounded by uncertainty, from potential manufacturing flaws, to delays, to losses, to mere natural randomness. Hence, extending Declare (and by extension LTL$_f$) with uncertainty expressions is necessary.

While probabilities are a natural candidate formalism for dealing with uncertainty, they are notoriously difficult to manage both in terms of modelling—since the lack of truth functionality requires the specification of a joint probability distribution or independence assumptions—and in terms of computational cost arising from the need to find all possible outcomes that define an event. Possibility measures thus arise as an alternative way to handle uncertainty. The main difference between possibilities and probabilities is that the former focuses more on a *qualitative* approach, where the specific degrees are less important than their ordering.

Our first proposal, ΠLTL$_f^0$, can be seen as a backbone for a possibilistic variant of Declare. If we think of a possibilistic Declare specification as a set of Declare constraints associated to possibility bounds, then it is equivalent to a ΠLTL$_f^0$ KB. Thus, our proposal is, to-date, the only formalism dealing with possibilistic Declare specifications. The limitation of this formalism is that it can only specify uncertainty about a full trace: the uncertainty is given, and managed, at the beginning of the execution.

In reality, however, uncertainty may depend on other events which have already been observed. For instance, in an online shop specification, the uncertainty of a product being delivered changes depending on whether it was already bought, packaged, or given to the courier. With full ΠLTL$_f$ it is possible to specify these distinctions, along many other complex constraints.

In this paper, we focused on the main reasoning task of deciding consistency of ΠLTL$_f$ and ΠLTL$_f^0$ KBs. In terms of process modelling, this corresponds to deciding whether a specification is realisable. We provided effective algorithms for answering this question. In the process modeling community it is also important to consider *monitoring*; that is, controlling that a current execution of a process does not lead to an unwanted scenario or a violation of the specification [20]. In future work, we plan to study this problem also for ΠLTL$_f$, following the approach proposed in [1]. We will also try to close the complexity gap for ΠLTL$_f$ consistency.

References

[1] A. Alman, F. M. Maggi, M. Montali, and R. Peñaloza. Probabilistic declarative process mining. *Inf. Syst.*, 109:102033, 2022. doi: 10.1016/J.IS.2022.102033.

[2] F. Baader, J. Hladik, and R. Peñaloza. Automata can show PSpace results for description logics. *Inf. Comput.*, 206(9-10):1045–1056, 2008.

[3] C. Baier and J.-P. Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[4] P. Billingsley. *Probability and measure*. A Wiley-Interscience publication. Wiley, New York, 3. ed edition, 1995. ISBN 0471007102.

[5] F. Chiariello, V. Fionda, A. Ielo, and F. Ricca. A direct ASP encoding for Declare. In M. Gebser and I. Sergey, editors, *Practical Aspects of Declarative Languages*, pages 116–133, Cham, 2023. Springer Nature Switzerland.

[6] M. Chodil and A. Kučera. The finite satisfiability problem for PCTL is undecidable. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '24. Association for Computing Machinery, 2024. doi: 10.1145/3661814.3662145.

[7] E. M. Clarke and I. A. Draghicescu. Expressibility results for linear-time and branching-time logics. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, pages 428–437, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[8] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In D. Kozen, editor, *Logics of Programs*, pages 52–71, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[9] G. De Giacomo and M. Y. Vardi. Linear temporal logic and linear dy-

namic logic on finite traces. In *Proceedings of IJCAI 2013*, pages 854–860. AAAI Press, 2013.

[10] D. Doder and A. Perović. Probabilistic temporal logics. In Z. Ognjanović, editor, *Probabilistic Extensions of Various Logical Systems*, pages 71–108. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-52954-3_3.

[11] D. Dubois and H. Prade. Possibility theory. In *Granular, Fuzzy, and Soft Computing*, pages 859–876. Springer, 2023.

[12] R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, USA, 4th edition, 2010. ISBN 0521765390.

[13] E. A. Emerson and J. Y. Halpern. "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *J. ACM*, 33(1):151–178, 1986. doi: 10.1145/4904.4999.

[14] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Form. Asp. Comput.*, 6(5):512–535, sep 1994. ISSN 0934-5043. doi: 10.1007/BF01211866.

[15] J. Hladik and U. Sattler. A translation of looping alternating automata into description logics. In F. Baader, editor, *Proceedings of CADE-19*, pages 90–105, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[16] S. Konur. Real-time and probabilistic temporal logics: An overview. *CoRR*, abs/1005.3200, 2010. URL http://arxiv.org/abs/1005.3200.

[17] A. Kovtunova and R. Peñaloza. Cutting diamonds: A temporal logic with probabilistic distributions. In *Proceedings of KR 2018*, pages 561–570. AAAI Press, 2018.

[18] Y. Li. Quantitative model checking of linear-time properties based on generalized possibility measures. *Fuzzy Sets and Systems*, 320:17–39, 2017. ISSN 0165-0114. doi: https://doi.org/10.1016/j.fss.2017.03.012. Theme Logic and Algebra.

[19] Y. Li, Y. Li, and Z. Ma. Computation tree logic model checking based on possibility measures. *Fuzzy Sets and Systems*, 262:44–59, 2015. ISSN 0165-0114. doi: https://doi.org/10.1016/j.fss.2014.03.009. Theme: Logic and Computer Science.

[20] F. M. Maggi, M. Montali, M. Westergaard, and W. M. P. van der Aalst. Monitoring business constraints with linear temporal logic: An approach based on colored automata. In *Proc. of BPM'11*, LNCS, pages 132–147. Springer, 2011.

[21] F. M. Maggi, M. Montali, and R. Peñaloza. Temporal logics over finite traces with uncertainty. In *Proceedings of AAAI 2020*, pages 10218–10225. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6583.

[22] F. M. Maggi, M. Montali, R. Peñaloza, and A. Alman. Extending temporal business constraints with uncertainty. In *Proc. of BPM 2020*, volume 12168 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2020. doi: 10.1007/978-3-030-58666-9_3.

[23] F. M. Maggi, M. Montali, and R. Peñaloza. Probabilistic temporal reasoning using superposition semantics. *ACM Trans. Comput. Log.*, 26 (2):9:1–9:26, 2025. doi: 10.1145/3714427.

[24] S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge Mathematical Library. Cambridge University Press, 2009. URL https://books.google.it/books?id=Md7RnYEPkJwC.

[25] L. Morão. Probabilistic distributed temporal logic. Master thesis, Universidade Técnica de Lisboa, Portugal, 2011.

[26] Z. Ognjanovic. Discrete linear-time probabilistic logics: Completeness, decidability and complexity. *J. Log. Comput.*, 16(2):257–285, 2006. doi: 10.1093/logcom/exi077.

[27] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst. Declare: Full support for loosely-structured processes. In *Proceedings of EDOC 2007*, pages 287–300. IEEE Computer Society, 2007.

[28] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pages 46–57, 1977. doi: 10.1109/SFCS.1977.32.

[29] R. Roy, J. Gaglione, N. Baharisangari, D. Neider, Z. Xu, and U. Topcu. Learning interpretable temporal properties from positive examples only. In B. Williams, Y. Chen, and J. Neville, editors, *Proceedings of AAAI 2023*, pages 6507–6515. AAAI Press, 2023. doi: 10.1609/AAAI.V37I5.25800.

[30] W. M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 1st edition, 2011. ISBN 3642193447.

[31] M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32(2):183–221, 1986. doi: 10.1016/0022-0000(86)90026-7.