Characterizing LTL Formulas by Examples

Balder ten Cate
ILLC, University of Amsterdam, The Netherlands

Dana Fisman
Institute for the Theory of Computing, Stein Faculty of Computer and Information Science, Ben Gurion University, Beer Sheva, Israel

Roi Ohayon
Institute for the Theory of Computing, Stein Faculty of Computer and Information Science, Ben Gurion University, Beer Sheva, Israel

Patrik Sestic
Unaffiliated researcher, Munich, Germany

Abstract

We investigate the extent to which Linear Temporal Logic (LTL) formulas can be uniquely characterized by a finite set of labeled examples. We consider different types of examples, ranging from finite words to transfinite words, as well as schematic examples. In the finite-word setting, we provide a complete classification of basis-restricted LTL fragments that admit such unique characterizations. Next, we show that allowing transfinite words as examples enables finite unique characterizations for large monotone fragments of LTL. Finally, we introduce schematic examples, i.e., patterns that compactly represent a family of finite words, and we show that these enable unique characterization results in the finite setting that were not possible with ordinary finite examples alone. Overall, the work provides a foundational account of the descriptive power of different example types for example-driven specification, debugging, and learning of temporal properties.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Linear Temporal Logic, Examples, Transfinite Words

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.19

Related Version *Full Version*: `https://arxiv.org/abs/2604.22097` [30]

Funding *Roi Ohayon*: Supported by ISF grant 250721.

Acknowledgements We thank Frank Wolter and Raoul Koudijs for their input on early versions.

1 Introduction

Linear Temporal Logic (LTL) [26] is widely used in formal verification, as a language for specifying requirements on the intended behavior of systems. In practice, writing LTL formulas is an error-prone process. Tools often rely on examples, i.e., concrete words (traces) that either satisfy or violate the specification, because they are easy to inspect and communicate. Examples arise naturally in automata-theoretic model checking: when a finite-state system violates an LTL property, one can extract a finite or an ultimately periodic counterexample [31]. Considerable work studies how to produce useful counterexamples, including short counterexamples [28]. Examples are also used as an interface for understanding and debugging temporal specifications [1]. Complementarily, there is a growing literature on learning temporal specifications from labeled examples [25, 6], and on interactive workflows that refine a specification by querying the user with candidate behaviors [15].

In this paper, we study a foundational question that underlies the above work: how much information about an LTL-formula can be conveyed by a finite set of labeled examples? More specifically, *when does there exist a finite set of labeled examples that uniquely identifies a* *formula (up to semantic equivalence)?* We adopt a learner-independent notion of identifiability: a set of labeled examples *uniquely characterizes* a target formula within a formula class if no other formula in the class fits it. This is the classical notion of a *teaching set* from computational learning theory [17]. We study the existence of such uniquely characterizing sets of examples, for different types of examples, including *finite words*, *ω-words*, *transfinite words*, and *schematic examples*. These different types of examples turn out to differ in their descriptive power when it comes to distinguishing LTL formulas.

▶ **Example 1.1.** The LTL-formula $\mathtt{F}p$ is distinguished from $\bigvee_{i\leq n}\mathtt{X}^i p$ by the example $\emptyset^{n+2}\{p\}$. The same formula cannot, however, be distinguished from all formulas of the form $\bigvee_{i\leq n}\mathtt{X}^i p$ using only finitely many labeled examples, if the examples are finite words, or even ω-words. On the other hand, the transfinite word $\emptyset^\omega\{p\}$, of length $\omega+1$, satisfies $\mathtt{F}p$ while falsifying $\bigvee_{i\leq n}\mathtt{X}^i p$ for all n. Similarly, the schematic example $\emptyset^*\{p\}$ is a positive example for $\mathtt{F}p$, in the sense that all instances of the schema are words satisfying $\mathtt{F}p$, and it is not a positive example in the same sense for $\bigvee_{i\leq n}\mathtt{X}^i p$ for any n.

We start from the finite-words setting. Building on prior work [13], we show that, in this setting, unique characterizability is the exception rather than the rule: for most fragments of LTL, over finite words, no finite set of positive/negative examples can uniquely characterize a formula up to equivalence. Moving from finite words to ω-words improves the picture, but the improvement is limited. A key shift comes when we go beyond ω-words and consider transfinite words as examples. Transfinite ordinal-indexed words allow us to separate formulas in a more robust way, yielding stronger positive results and, in particular, existence of finite uniquely characterizing sets for meaningful fragments of LTL. Finally, we use these transfinite examples as a stepping stone towards new results in the finite setting: we introduce *schematic examples*, i.e., finite patterns that stand for a family of concrete examples. We show how the additional power available in the transfinite setting can be compiled into such patterns. This yields unique characterization results that are unattainable with concrete finite examples alone, while still keeping the examples close to the finite descriptions familiar to practitioners.

More concretely, our main contributions are:

1. In the finite-word setting, we provide a complete classification of fragments LTL_O with $O\subseteq\{\mathtt{U},\mathtt{F},\hat{\mathtt{F}},\mathtt{X},\wedge,\vee,\neg,\top,\bot\}$, with respect to whether the fragment in question admits finite unique characterizations when interpreted over finite words. There are six maximal such fragments that do admit finite unique characterizations, but they are rather limited in expressive power. In particular, as was observed already in [13], even $\mathsf{LTL}_{\mathtt{F},\wedge}$ already does not admit finite characterizations, and neither does $\mathsf{LTL}_{\mathtt{X},\vee}$.
2. Moving on to ω-words, we show that, although $\mathsf{LTL}_{\mathtt{X},\wedge,\vee,\top,\bot}$ admits finite characterizations consisting of ω-words, other fragments such as $\mathsf{LTL}_{\mathtt{F},\wedge}$ and $\mathsf{LTL}_{\mathtt{G},\vee}$ still do not.
3. We show that $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$ admits finite characterizations consisting of (finitely presentable) transfinite words. A similar result holds for the dual fragment $\mathsf{LTL}_{\hat{\mathtt{G}},\wedge,\vee,\top,\bot}$.
4. Returning to the finite-word setting, we introduce the concept of *schematic examples* and we show that $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$ admits finite characterizations by schematic examples.

Due to space constraints, most proofs are deferred to the full version [30]. A python library and an interactive demonstration of the concepts and algorithms developed in this paper can be found at `https://ltl-workbench.onrender.com/`.

Related work

The problem of inferring temporal specifications from examples has attracted considerable attention in recent years. SAT-based and decision-tree-based approaches have been proposed for learning LTL formulas from positive and negative examples, aiming to find a small formula consistent with the given sample [25]. This task is often described as separating positive from negative examples; in our terminology, this means finding a formula that *fits* the sample. Learning LTL over finite traces (LTLf) is considered in [27]. An approach combining passive learning with syntactic constraints on the space of inferred formulas is pursued in [34]. A different direction studies whether neural networks can learn semantic aspects of temporal logic, such as producing satisfying traces for LTL formulas [18].

Our work is related to passive learning from examples, but asks a different question. Passive learning asks whether, given a sample, there exists a formula in a given class that fits it, often seeking a small such formula. In contrast, we study when a given LTL formula can be *uniquely characterized* by a finite set of labeled examples, relative to a class of formulas. Thus, our focus is not on synthesizing one consistent explanation, but on understanding which fragments admit finite samples that rule out all inequivalent alternatives.

A more foundational line of work studies the computational complexity of passive learning. It is shown that passive learning of LTL formulas is NP-complete for almost all standard fragments [23], and this is extended to LTL, CTL, and ATL, where the problem remains NP-complete with unbounded binary operators, while bounded-binary fragments exhibit more fine-grained differences [4]. These works identify fundamental algorithmic limits of example-based specification: given a sample, decide whether some formula in the class fits it. In contrast, our focus is semantic: given a fragment, we determine whether every formula admits a finite sample that uniquely identifies it up to equivalence.

A related perspective is that of machine teaching. Adaptive teaching of parametric LTL (pLTL) formulas to learners with preferences is studied in [33].[1] This line of work is algorithmic and learner-dependent: examples are chosen with respect to a given learner model and preference structure. In contrast, our notion of characterizability is learner-independent: a characterizing set must identify the target formula semantically among all formulas in the class, independently of any learning algorithm or preference relation. Thus, our results isolate structural limits of example-based teaching for LTL.

Another related thread aims to explain LTL formulas using words or abstractions thereof. For example, [21] presents visualizations of an LTL formula's behavior in a form close to the "timeline sketches" used by practitioners; see also [32]. While such work focuses on making a given formula more understandable, our goal is to understand when examples themselves determine the formula.

In the knowledge representation literature, motivated by exact learnability questions in active-learning settings with membership queries, the question has been studied whether a description logic concept is uniquely characterized by a finite set of positive and negative examples [29, 14]. This study was further extended to temporalized description logic concepts in [13, 19]. In this context, [13] also studied unique characterizations for fragments of LTL over finite words. We will build on some of the results from [13].

LTL over transfinite words was proposed as a means to represent systems with Zeno-like behaviors and multi-phase executions in [8, 9], where it was shown that the usual automata-theoretic techniques for LTL can be extended suitably to transfinite words, yielding satisfiability and model-checking procedures with similar complexity as over ω-words.

[1] In pLTL, the F and G operators are associated with a parameter bounding their number of time steps.

2 Preliminaries

Transfinite words

We briefly recall standard notions on ordinals needed in the sequel. An ordinal is a well-ordered set, where every non-empty subset admits a least element, and ordinals are identified up to order isomorphism. Every ordinal is either 0, a successor ordinal of the form $\beta + 1$, or a limit ordinal. Ordinal arithmetic (addition, multiplication, and exponentiation) is defined inductively and is, in general, non-commutative. In particular, for any ordinals $\alpha < \beta$ there exists a unique ordinal γ such that $\beta = \alpha + \gamma$, and for any α and $\beta > 0$ there exist unique ordinals γ and $\delta < \beta$ such that $\alpha = \beta \cdot \gamma + \delta$. We refer to standard texts (e.g., Jech (1997)) for further details. We will only be concerned with *countable* ordinals, i.e., ordinals α such that $\{\beta \mid \beta < \alpha\}$ is a countable set.

▶ **Definition 2.1** (α-words). *Fix a finite set of atomic propositions* AP*. If* α *is a countable ordinal, then an* α-word *is a function* $w : \alpha \to \mathcal{P}(\mathit{AP})$*.*

We think of an α-word w as a (possibly transfinite) sequence of valuations $w[0]\,w[1]\,w[2]\,\ldots$, indexed by the ordinals below α, where $w[\beta]$ is the valuation at position β. If $\alpha > \omega$, the sequence extends beyond the finite positions. We refer to α-words as *ordinal words.* If $\alpha \geq \omega$, we also refer to them as *transfinite words.*

We also write $|w|$ to denote the ordinal α. For $\beta < |w|$, we write $w[\beta..]$ to denote the suffix of w starting at position β. Formally, $w[\beta..]$ is the $(|w| - \beta)$-word w' such that $w'[\gamma] = w[\beta + \gamma]$.

For a countable ordinal α we denote by $\mathcal{W}^{\leq\alpha}$ the set of all β-words w with $\beta \leq \alpha$. Similarly, we denote by $\mathcal{W}^{<\alpha}$ and $\mathcal{W}^{=\alpha}$ the set of all ordinal words of length less than α, respectively, of length exactly α. We will also write $\mathcal{W}^{\mathsf{fin}}$ for $\mathcal{W}^{<\omega}$ and $\mathcal{W}^{\mathsf{all}}$ for the set of all countable ordinal words.

Although α-words do not necessarily follow a repeating pattern, we will be particularly interested in α-words that do, since, despite their possibly infinite length, they can be represented by finite expressions. We introduce the following word-expression language for their representation.

▶ **Definition 2.2** (Word-expressions). *A* word-expression *over* AP *is any expression generated inductively by the following rules:*

- *Each* $\sigma \subseteq \mathit{AP}$ *constitutes a* 1*-word-expression. It denotes the* 1*-word* w_σ *where* $w_\sigma[0] = \sigma$*.*
- *If* A *is an* α*-word-expression and* B *is a* β*-word-expression, then* $A \cdot B$ *is an* $(\alpha + \beta)$*-word-expression. It denotes the* $(\alpha + \beta)$*-word* $w_{A \cdot B}$ *defined by*

$$w_{A\cdot B}[\gamma] = \begin{cases} w_A[\gamma] & \textit{if } \gamma \in \alpha \\ w_B[\gamma - \alpha], & \textit{otherwise.} \end{cases}$$

  We sometimes omit the concatenation symbol $\cdot$ *when no ambiguity arises.*
- *If* A *is an* α*-word-expression, then* $(A)^\omega$ *is an* $(\alpha \cdot \omega)$*-word-expression. It denotes the* $(\alpha \cdot \omega)$*-word* $w_{(A)^\omega}$ *defined, for all* $\gamma < \alpha \cdot \omega$*, by* $w_{(A)^\omega}[\gamma] = w_A[\gamma \bmod \alpha]$*.*

We say that an α-word is *finitely presentable* (or, *regular*) if there is an α-word expression that denotes it.

▶ **Example 2.3.** The word-expression $\{q\}^\omega \cdot \{p\} \cdot \{q\}^\omega$ denotes a transfinite word of length $\omega + \omega$ that can be depicted as follows: $\{q\} \longrightarrow \{q\} \longrightarrow \ldots \;\; \{p\} \longrightarrow \{q\} \longrightarrow \{q\} \longrightarrow \ldots.$

Table 1 Semantics of LTL over ordinal words.

$w \models \top$		$w \models \mathtt{X}\varphi$	iff $\lvert w\rvert > 1$ and $w[1..] \models \varphi$
$w \models p$	iff $p \in w[0]$	$w \models \mathtt{F}\varphi$	iff $\exists \alpha < \lvert w\rvert$ s.t. $w[\alpha..] \models \varphi$
$w \models \neg\varphi$	iff $w \not\models \varphi$	$w \models \varphi \,\mathtt{U}\, \psi$	iff $\exists \gamma$ with $0 \leq \gamma < \lvert w\rvert$ s.t. $w[\gamma..] \models \psi$ and $\forall \beta$ with $0 \leq \beta < \gamma$, $w[\beta..] \models \varphi$
$w \models \varphi \wedge \psi$	iff $w \models \varphi$ and $w \models \psi$		

We note that, although the definitions above apply to arbitrary countable ordinals, all transfinite words considered in this paper have length below ω^2. In particular, they contain only finitely many infinite stretches.

Linear Temporal Logic

Let $\mathtt{AP}$ be a set of atomic propositions. An *LTL*[*AP*] *formula* is any expression generated by the following grammar:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathtt{X}\varphi \mid \mathtt{F}\varphi \mid \varphi_1 \,\mathtt{U}\, \varphi_2$$

We can define some other logical connectives and temporal operators as abbreviations:

$$\bot \;:=\; \neg\top \qquad \begin{array}{lcl} \varphi_1 \vee \varphi_2 & := & \neg(\neg\varphi_1 \wedge \neg\varphi_2) \\ \psi_1 \rightarrow \psi_2 & := & \neg\psi_1 \vee \psi_2 \end{array} \qquad \begin{array}{lcl} \hat{\mathtt{F}}\psi & := & \mathtt{XF}\psi \\ \mathtt{G}\psi & := & \neg\mathtt{F}\neg\psi \end{array}$$

We will also use $\mathtt{X}^n\varphi$ as shorthand for $\mathtt{X}\cdots\mathtt{X}\varphi$ with n occurrences of $\mathtt{X}$, and similarly for $\hat{\mathtt{F}}^n\varphi$.

We will be interested in syntactic fragments of LTL. For $O \subseteq \{\mathtt{U}, \mathtt{F}, \hat{\mathtt{F}}, \mathtt{X}, \wedge, \vee, \neg, \top, \bot\}$, let LTL_O denote the fragment of LTL consisting of formulas built from atomic propositions using only the connectives in O. We write $\mathsf{LTL}_O[\mathtt{AP}]$ for the restriction of LTL_O to formulas over the set of propositions $\mathtt{AP}$.

The semantics of LTL, over arbitrary ordinal words w, is given in Table 1.

We will denote by $[\![\varphi]\!]$ the set of all ordinal words w such that $w \models \varphi$. We will also write $[\![\varphi]\!]^{\text{fin}}$ for $[\![\varphi]\!] \cap \mathcal{W}^{\mathsf{fin}}$, and, for all ordinals α, $[\![\varphi]\!]^{\leq\alpha}$, $[\![\varphi]\!]^{<\alpha}$ and $[\![\varphi]\!]^{=\alpha}$ are defined similarly. Note that, whether two LTL-formulas are equivalent depends on the class of words over which they are interpreted. This is illustrated by the following examples.

▶ **Example 2.4.** $[\![\mathtt{FX}p]\!]^{\leq\omega} = [\![\mathtt{XF}p]\!]^{\leq\omega}$ but $[\![\mathtt{FX}p]\!] \neq [\![\mathtt{XF}p]\!]$, as witnessed by the $\omega+\omega$-length word in Example 2.3, which satisfies $\mathtt{XF}p$ and not $\mathtt{FX}p$.

▶ **Example 2.5.** Recall $\mathtt{G}\varphi$ is shorthand for $\neg\mathtt{F}\neg\varphi$. $[\![\mathtt{GX}p]\!]^{\text{fin}} = [\![\bot]\!]^{\text{fin}}$ but $[\![\mathtt{GX}p]\!]^{=\omega} \neq [\![\bot]\!]^{=\omega}$.

On the other hand, for the fragment $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$, there is no difference between finite words and arbitrary ordinal words when it comes to equivalence:

▶ **Theorem 2.6.** *Let φ, ψ be $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$-formulas. Then $[\![\varphi]\!] = [\![\psi]\!]$ iff $[\![\varphi]\!]^{\text{fin}} = [\![\psi]\!]^{\text{fin}}$. The same holds for $\mathsf{LTL}_{\mathtt{X},\wedge,\neg}$.*

For LTL over transfinite words, one can restrict attention to finitely presentable words:

▶ **Theorem 2.7.** *Let φ, ψ be LTL-formulas. If the set $[\![\varphi]\!] \setminus [\![\psi]\!]$ is non-empty, it contains a finitely presentable word.*

3 Characterizing Formulas by Examples

By a *labeled example* we will mean a pair (w, lab) where w is an ordinal word and $lab \in \{+, -\}$. An LTL-formula φ *fits* a labeled example (w, lab) if either $w \models \varphi$ and $lab = +$, or $w \not\models \varphi$ and $lab = -$. A set of labeled examples E *uniquely characterizes* an LTL-formula φ relative to $\mathsf{LTL}_O[\mathtt{AP}]$ (for some set of operators O and some set of atomic propositions $\mathtt{AP}$) if (i) φ fits every labeled example in E, and (ii) for every $\mathsf{LTL}_O[\mathtt{AP}]$-formula ψ that fits all labeled examples in E, we have that $[\![\psi]\!] = [\![\varphi]\!]$. Uniquely characterizing sets of examples are also known as *teaching sets* in the literature. We say that $\mathsf{LTL}_O[\mathtt{AP}]$ *admits finite characterizations* if every $\varphi \in \mathsf{LTL}_O[\mathtt{AP}]$ is uniquely characterized, relative to $\mathsf{LTL}_O[\mathtt{AP}]$, by a finite set of labeled examples.

▶ **Example 3.1.** Consider the formula $\varphi = p \wedge \mathtt{X}p$. This formula fits the following labeled examples: $(\{p\}\{p\}, +)$, $(\emptyset\{p\}, -)$, and $(\{p\}\emptyset, -)$. Moreover, φ is uniquely characterized, relative to $\mathsf{LTL}_{\mathtt{X},\wedge}[\{p, q\}]$, by these labeled examples. On the other hand, φ is *not* uniquely characterized by these examples relative to $\mathsf{LTL}_{\mathtt{X},\wedge,\vee}[\{p, q\}]$, since the formula $\psi = \varphi \vee q$ also fits and $[\![\varphi]\!] \neq [\![\psi]\!]$.

Observe that positive results are preserved under restricting the class of formulas, whereas negative results are preserved under enlarging it. Thus, every positive result for $\mathsf{LTL}_O[\mathtt{AP}]$ also holds for $\mathsf{LTL}_{O'}[\mathtt{AP}']$ whenever $O' \subseteq O$ and $|\mathtt{AP}'| \leq |\mathtt{AP}|$. Conversely, every negative result for $\mathsf{LTL}_O[\mathtt{AP}]$ also holds for $\mathsf{LTL}_{O'}[\mathtt{AP}']$ whenever $O \subseteq O'$ and $|\mathtt{AP}| \leq |\mathtt{AP}'|$.

These notions can be further refined by restricting attention to words of certain lengths (e.g., finite words, or words of length at most ω). We say that $\mathsf{LTL}_O[\mathtt{AP}]$ admits finite characterizations *over* $\mathcal{W}$ if for every $\varphi \in \mathsf{LTL}_O[\mathtt{AP}]$ there is a finite set of labeled examples E consisting of words in $\mathcal{W}$, such that (i) φ fits every labeled example in E, and (ii) for every $\mathsf{LTL}_O[\mathtt{AP}]$-formula ψ that fits all labeled examples in E, we have that $[\![\psi]\!] \cap \mathcal{W} = [\![\varphi]\!] \cap \mathcal{W}$.

If an LTL-formula φ admits a uniquely characterizing set of labeled examples E with respect to a fragment, it means that *every* consistent learner for the given fragment is guaranteed to output φ (or an equivalent formula) by including the examples in E in the input. A weaker, learner-dependent notion of *characteristic samples* will be discussed in Section 8.

4 Results for LTL over finite words

Fortin et al. [13] previously studied the existence of finite unique characterizations for fragments of LTL over finite words. In particular, they showed that $\mathsf{LTL}_{\mathtt{X},\hat{\mathtt{F}},\wedge,\top}[\mathtt{AP}]$ admits finite characterizations over $\mathcal{W}^{\mathsf{fin}}$. On the other hand, they showed that $\mathsf{LTL}_{\mathtt{F},\wedge}[\{p, q\}]$ and $\mathsf{LTL}_{\mathtt{U},\mathtt{X}}[\{p\}]$ do not admit finite characterizations over $\mathcal{W}^{\mathsf{fin}}$. [2] They also consider subfragments consisting of *path-shaped* formulas, in which conjunctions of temporal statements (such as $\mathtt{F}p \wedge \mathtt{F}q$) are disallowed. For instance, they showed that even the path-shaped subfragment of $\mathsf{LTL}_{\mathtt{F},\wedge}[\{p, q\}]$ does not admit finite characterizations.

▶ **Example 4.1** ([13]). $\mathsf{LTL}_{\mathtt{F},\wedge}[\{p, q\}]$ does not admit finite characterizations over $\mathcal{W}^{\mathsf{fin}}$, and this holds already even just for path-shaped formulas. Specifically, the formula $\mathtt{F}(p \wedge q)$ cannot be distinguished from all formulas of the form $\mathtt{F}(p \wedge \mathtt{F}(q \wedge \mathtt{F}(p \wedge \mathtt{F}(q \wedge \cdots))))$ using only finitely many examples, where examples are finite words.

[2] A stronger positive result is stated in [13] pertaining to $\mathsf{LTL}_{\hat{\mathtt{F}},\mathtt{X},\wedge,\top,\bot}[\mathtt{AP}]$, but the proof is flawed and applies only to the fragment without $\bot$. Also, note that [13] uses a different variant of the Until operator.

In [13], the authors subsequently perform an in-depth investigation into the existence of finite characterizations for various path-shaped subfragments of $\mathsf{LTL}_{\{\mathtt{F},\hat{\mathtt{F}},\mathtt{X},\wedge,\top,\bot\}}[\mathtt{AP}]$. We will not consider such path-shaped fragments here, since they are quite restrictive, and we refer the reader to [13] for details.

The aforementioned results raise the question whether there may be other natural syntactic fragments of LTL that admit finite characterizations over $\mathcal{W}^{\mathsf{fin}}$. The following theorem provides an exhaustive answer to this question, for fragments obtained by restricting the set of allowed Boolean and temporal connectives. The proof is by an exhaustive case analysis.

▶ **Theorem 4.2.** *Let AP be any finite set of atomic propositions with $|AP| \geq 2$, and let $O \subseteq \{U, F, \hat{F}, X, \wedge, \vee, \neg, \top, \bot\}$ containing at least one temporal operator. Then $\mathsf{LTL}_O[AP]$ admits finite characterizations over $\mathcal{W}^{\mathsf{fin}}$ if and only if O is a subset of one of the following sets:*

1. $\{\hat{F}, X, \wedge, \top\}$
2. $\{F, \vee, \top, \bot\}$
3. $\{F, \neg, \top, \bot\}$
4. $\{F, \hat{F}, X, \top\}$
5. $\{X, \neg\}$
6. $\{\hat{F}, \neg\}$

A few observations are worth making, based on the above theorem. First of all, any fragment of LTL containing the $\mathtt{U}$ operator fails to admit finite characterizations. Secondly, fragments of LTL that include all Boolean connectives (and at least one temporal operator) do not admit finite characterizations. In fact, in order to admit finite characterizations, a fragment of LTL must omit conjunction or disjunction. However, omitting conjunction or disjunction is not enough: $\mathsf{LTL}_{\mathtt{F},\wedge}$ does not admit finite characterizations over $\mathcal{W}^{\mathsf{fin}}$ (as we already discussed) and the same holds for $\mathsf{LTL}_{\hat{\mathtt{F}},\vee}$ and $\mathsf{LTL}_{\mathtt{X},\vee}$.

▶ **Example 4.3.** $\mathsf{LTL}_{\mathtt{X},\vee}[\{p\}]$ does not admit finite characterizations over $\mathcal{W}^{\mathsf{fin}}$. Indeed, the formula p cannot be distinguished from all formulas of the form $p \vee \mathtt{X}^n p$ $(n > 0)$ using only finitely many labeled examples, where the examples are finite words. A similar argument applies to $\mathsf{LTL}_{\hat{\mathtt{F}},\vee}[\{p\}]$.

As we will see in the next sections, omitting negation as a Boolean connective *can* be enough to obtain finite characterizability once we go beyond finite words.

5 Results for LTL over ω-words

We start by giving an example showing that infinite words can help when it comes to characterizing LTL-formulas.

▶ **Example 5.1.** We saw in Example 4.1 that $\mathtt{F}(p \wedge q)$ is not finitely characterizable in $\mathsf{LTL}_{\mathtt{F},\wedge}[\{p,q\}]$ over $\mathcal{W}^{\mathsf{fin}}$. This formula *is* finitely characterizable in $\mathsf{LTL}_{\mathtt{F},\wedge,\vee,\top,\bot}[\{p,q\}]$ (over $\mathcal{W}^{\mathsf{all}}$) by finitely presentable ω-words, namely by the positive example $\emptyset\ \{p,q\}\ \emptyset^\omega$ (or simply $\emptyset\ \{p,q\}$) and the negative example $(\{p\}\ \{q\})^\omega$. On the other hand, the formula $\mathtt{F}(p \wedge q \wedge \mathtt{F}(r \wedge \mathtt{F}(p \wedge q)))$ is *not* finitely characterizable by ω-words (cf. the proof of Theorem 5.3 below).

This may suggest that, by allowing ω-words, we can circumvent the negative results from the previous section. This is partly true. For example, we have:

▶ **Theorem 5.2.** *Let AP be any finite set of atomic propositions. $\mathsf{LTL}_{X,\wedge,\vee,\top,\bot}[AP]$ admits finite characterizations over $\mathcal{W}^{\leq\omega}$ and over $\mathcal{W}^{=\omega}$. In fact, every $\mathsf{LTL}_{X,\wedge,\vee,\top,\bot}[AP]$-formula is uniquely characterized with respect to $\mathsf{LTL}_{X,\wedge,\vee,\top,\bot}[AP]$ (over $\mathcal{W}^{\mathsf{all}}$) by a finite set of finitely presentable labeled examples of ordinal length at most ω.*

The proof uses a monotonicity argument to show that, with finitely many labeled examples, we can force any fitting formula to depend only on a finite fixed-length prefix.

Unfortunately, other fragments of LTL still do not admit finite characterizations, even when ω-words are allowed. In particular,

▶ **Theorem 5.3.** *$\mathsf{LTL}_{\mathtt{F},\wedge}[\{p,q,r\}]$ does not admit finite characterizations over $\mathcal{W}^{\leq\omega}$ or $\mathcal{W}^{=\omega}$.*

Since $\mathtt{F}\varphi$ is definable as $\varphi \vee \hat{\mathtt{F}}\varphi$, the same holds for $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee}[\{p,q,r\}]$. We will refrain from providing a complete classification in the style of Theorem 4.2. Instead, we will next move on to the setting where examples can be transfinite words – a setting where, as we will see, more positive results hold.

6 Results for LTL over transfinite words

Our main result, in this section, is:

▶ **Theorem 6.1.** *Let $\mathtt{AP}$ be any finite set of atomic propositions. Then $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}[\mathtt{AP}]$ admits finite characterizations (over $\mathcal{W}^{all}$). In fact, every formula in this fragment is uniquely characterized by a finite set of finitely presentable labeled examples of length less than ω^2.*

Note that, since $\mathtt{F}\varphi$ is definable as $\varphi\vee\hat{\mathtt{F}}\varphi$, the result extends immediately to $\mathsf{LTL}_{\mathtt{F},\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$.

▶ **Example 6.2.** Consider again the formula $\mathtt{F}(p \wedge q \wedge \mathtt{F}(r \wedge \mathtt{F}(p \wedge q)))$. We saw earlier that this formula cannot be uniquely characterized w.r.t. $\mathsf{LTL}_{\mathtt{F},\wedge}[\{p,q,r\}]$ over $\mathcal{W}^{\leq\omega}$ by finitely many labeled examples (of length at most ω). The above theorem tells us that the same formula *can* be uniquely characterized using transfinite examples. Indeed, it is uniquely characterized by $S = \big(E^+ \times \{+\},\ E^- \times \{-\}\big)$, where

$$E^+ = \{\{p,q\}\{r\}\{p,q\},\ \{p,q\}\{p,q,r\},\ \{p,q,r\},\ \emptyset\{p,q\}\{r\}\{p,q\},\ \emptyset\{p,q\}\{p,q,r\},\ \emptyset\{p,q,r\}\},$$

$$E^- = \{\{p,q\}^\omega\cdot(\{p,r\}\{q,r\})^\omega,\ (\{p,r\}\{q,r\})^\omega\cdot\{p,q\}^\omega\cdot(\{p,r\}\{q,r\})^\omega,\ (\{q,r\}\{p,r\})^\omega\cdot\{p,q\}^\omega\cdot(\{p,r\}\{q,r\})^\omega\}.$$

The examples above illustrate why words of length below ω^2 suffice. Intuitively, each level of nesting of the $\mathtt{F}$ operator requires at most one additional ω-segment in the negative examples. Since every formula has finite nesting depth, finitely many ω-segments always suffice.

Theorem 6.1 is optimal, as finite characterizations are lost when negation is added to the fragment.

▶ **Theorem 6.3.** *$\mathsf{LTL}_{\mathtt{X},\wedge,\neg}[\{p\}]$, $\mathsf{LTL}_{\mathtt{F},\wedge,\neg}[\{p\}]$ and $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\neg}[\{p\}]$ do not admit finite characterizations over $\mathcal{W}^{all}$.*

Moreover, extending the fragment $\mathsf{LTL}_{\mathtt{F},\hat{\mathtt{F}},\wedge,\vee,\top,\bot}$ with the $\mathtt{X}$ operator also destroys finite characterizability.

▶ **Theorem 6.4.** *$\mathsf{LTL}_{\mathtt{F},\mathtt{X},\wedge,\vee}[\mathtt{AP}]$ does not admit finite characterizations over $\mathcal{W}^{all}$.*

In the remainder of this section, we describe the proof of Theorem 6.1. It relies on viewing formulas through their minimal witnesses and how these witnesses propagate across words. To formalize this idea, we introduce a notion of *structural embedding* between words, captured by an order-preserving, label-monotone *homomorphism*.

▶ **Definition 6.5** (Homomorphism). *A* homomorphism *from an α-word w to a β-word w' is a function $h : \alpha \to \beta$ such that:*
1. $h(0) = 0$*;*
2. $\forall i < \alpha$, $w[i] \subseteq w'[h(i)]$*; and*
3. $\forall i < j < \alpha$, $h(i) < h(j)$*;*

When such a homomorphism exists, we write $w \leq_{\text{hom}} w'$. If condition (1) is dropped, we say that h is a *weak homomorphism* and write $w \leq_{\text{hom}^+} w'$. For a set of words E, we denote by $E^{\uparrow}$ the *upward closure* of E, i.e., $\{ w \mid \exists e \in E : e \leq_{\text{hom}} w \}$.

A central property of the fragment is that satisfaction is preserved under such embeddings. Intuitively, enlarging labels and stretching positions cannot destroy the truth of formulas built from $\mathtt{F}$, $\hat{\mathtt{F}}$, and Boolean connectives.

▶ **Lemma 6.6** (Monotonicity w.r.t. $\leq_{\text{hom}}$). *For every $\varphi \in \mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}[\mathit{AP}]$ and ordinal words w, w', if $w \models \varphi$ and $w \leq_{\text{hom}} w'$, then $w' \models \varphi$.*

Monotonicity suggests that formulas can be characterized by their minimal satisfying words. We now show that each formula admits a finite set of such minimal witnesses, from which all satisfying words can be obtained via homomorphic extension.

▶ **Lemma 6.7** (Canonical set). *For every $\varphi \in \mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}[\mathit{AP}]$, there exists a finite set $E^+_\varphi \subseteq (2^{\mathit{AP}})^{<\omega}$ such that $[\![\varphi]\!] = \{w \mid \exists e \in E^+_\varphi, e \leq_{hom} w\}$. We call E^+_φ the* canonical set *of φ.*

The proof of the above lemma is based on an inductive construction. The interesting case is for conjunctions, where $E^+_{\varphi\wedge\psi}$ must consist of homomorphically minimal examples satisfying both φ and ψ. Such a set can be constructed by taking, for each pair of words $w_1 \in E^+_\varphi$ and $w_2 \in E^+_\psi$, all possible interleaving of w_1 and w_2.

While the canonical set captures all positive behavior of a formula, it does not by itself yield a characterization: one must also ensure that words not satisfying the formula are excluded. To this end, we construct a finite family of maximal negative examples that block all possible embeddings of positive witnesses.

▶ **Lemma 6.8** (Existence of maximal negative examples). *For every $\varphi \in \mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}[\mathit{AP}]$, there exists a* finite *set $E^-_\varphi \subseteq (2^{\mathit{AP}})^{<\omega^2}$ such that*

$$\forall w \in (2^{\mathit{AP}})^{\leq\omega} \ \big(w \not\models \varphi \Rightarrow \exists f \in E^-_\varphi : w \leq_{\text{hom}} f\big), \qquad \textit{and} \qquad \forall f \in E^-_\varphi,\ f \not\models \varphi.$$

We call E^-_φ a set of maximal negative examples *for φ.*

Lemma 6.8 is proved by dualizing the canonical set E^+_φ. More precisely, we build a finite set of words that systematically block all possible homomorphic embeddings of examples in E^+_φ. A key point is that blocking an embedding requires accounting for all possible positions at which the letters of a positive witness may be matched. For instance, preventing an embedding of $e = \{p\}\{q\}\{p\}$ requires considering all cases in which $\{q\}$ is matched after an arbitrary finite prefix following $\{p\}$, and then ensuring that $\{p\}$ does not appear thereafter. Since the length of the prefix is unbounded, the construction must range over all finite lengths, and then enforce an additional infinite constraint. This can be done by a procedure that yields words of length up to ω^2.

Combining the canonical positive examples with the maximal negative examples yields a finite labeled sample that uniquely characterizes the formula, thereby proving Theorem 6.1.

By a further dualization argument we also obtain an analogous result for $\mathsf{LTL}_{\hat{\mathtt{G}},\wedge,\vee,\top,\bot}$:

▶ **Corollary 6.9.** *Let* AP *be any finite set of proposition letters.* $\mathsf{LTL}_{\hat{\mathtt{G}},\wedge,\vee,\top,\bot}[AP]$ *admits finite characterizations.*

▶ Remark 6.10. We can bound the size of the finite characterizations produced by our constructions. Let $m := |\varphi|$ denote the size of the syntax tree of φ, and let $h := |\mathtt{AP}|$. The construction of Lemmas 6.7 and 6.8 yields a finite characterization $S = (E^+_\varphi \times \{+\}, E^-_\varphi \times \{-\})$ such that $|E^+_\varphi| \leq 2^{O(m)}$ and $|E^-_\varphi| \leq 2^{O(m\,2^m)}$. Furthermore, each word is given by a word expression of size at most $O(mh2^{m+h})$. Thus, the overall size of the characterization is doubly exponential in $|\varphi|$. Proof details appear in the appendix. These bounds should be compared with the lower-bound results of Fortin et al. [13], who showed that $\mathsf{LTL}_{\wedge,\hat{\mathtt{F}}}$ does not admit polynomial characterizations over finite words.

7 Back to the Finite: Schematic Examples

We turn to define *schematic examples*, a concise way to represent (possibly) infinitely many finite examples. Consider a set of atomic propositions $\mathtt{AP}$. We use $\mathbb{B}(\mathtt{AP})$ to denote the Boolean expressions over $\mathtt{AP}$. Note that a Boolean expression $\mathtt{b} \in \mathbb{B}(\mathtt{AP})$ defines a set of letters over $\Sigma = 2^{\mathtt{AP}}$, the alphabet on which models of LTL are defined.

A *schematic example* is specified by a union-free regular expression r over the alphabet $\mathbb{B}(\mathtt{AP})$ of star-height at most 1 (that is, there are no nested occurrences of the Kleene star).We say that φ *fits* $(r, +)$ if $L(r) \subseteq [\![\varphi]\!]$, and that φ fits $(r, -)$ if $L(r) \cap [\![\varphi]\!] = \emptyset$. A schematic example can be viewed as an underspecified representation of a trace. This underspecification arises at two levels. First, each letter is a Boolean formula over the atomic propositions and therefore represents a set of possible valuations rather than a single, fully specified state. Second, Kleene star leaves the number of repetitions of a finite pattern unspecified, thereby abstracting from the exact duration of a phase or the number of times it occurs. We exclude union and restrict the star height to one, ensuring that every repeated subexpression is itself a fixed finite pattern containing no further unbounded repetition. Consequently, a schematic example can be read as a linear succession of finitely described phases, some of variable duration.

▶ **Example 7.1.** Fix $\mathtt{AP} = \{p, q\}$. The LTL-formula $\mathtt{F}(p \wedge q)$ is uniquely characterized w.r.t. $\mathsf{LTL}[\{p, q\}]$ over $\mathcal{W}^{\mathsf{fin}}$ by the set of labeled schematic examples

$$E = \{(\mathtt{true}^* \cdot (p \wedge q) \cdot \mathtt{true}^*), +), \quad ((\neg p \vee \neg q)^*, -)\} \qquad \text{where } \mathtt{true} = p \vee \neg p.$$

Building on the results about transfinite examples in the previous section, we can show:

▶ **Theorem 7.2.** $\mathsf{LTL}_{\hat{\mathtt{F}},\wedge,\vee,\top,\bot}[AP]$ *admits finite characterizations over* $\mathcal{W}^{\mathsf{fin}}$ *by schematic examples.*

Proof sketch. The construction proceeds by reducing finite characterizations consisting of regular (i.e., finitely presented) transfinite words to characterizations consisting of schematic examples.

We already showed that every formula φ in the fragment admits a finite characterization by a sample S consisting of positively labeled finite words and negatively labeled regular transfinite words. We transform this sample into a schematic sample $S' := \mathit{Schematic}(S)$ by replacing each word w with its schematic representation $r(w)$.

The key observation is that, for this fragment, satisfaction is preserved under replacing ω-iterations by finite repetitions. Thus each schematic expression $r(w)$ faithfully captures the behavior of the original word w with respect to φ. Consequently, S' fits φ, and moreover, any formula that fits S' must also fit S, and hence is equivalent to φ.

Finally, by construction, every schematic expression in S' is obtained by replacing ω-powers with Kleene-star, and therefore has star-height 1. ◀

▶ **Example 7.3** (Schematic examples)**.** Recall the formula $\varphi = \mathtt{F}\big(p \wedge q \wedge \mathtt{F}(r \wedge \mathtt{F}(p \wedge q))\big)$ from Example 6.2. We now give the corresponding schematic examples.

The positive examples can be grouped into two schematic examples:

$$R^+ = \{\mathtt{true}^* \cdot (p{\wedge}q{\wedge}r), \quad \mathtt{true}^* \cdot (p{\wedge}q) \cdot r \cdot (p{\wedge}q)\}.$$

The negative examples can be grouped into the single schematic example

$$R^- = \{(\neg(p{\wedge}q))^* \cdot (\neg r)^* \cdot (\neg(p{\wedge}q))^*\}.$$

Using $\mathbb{B}(\mathtt{AP})$ as the basis of regular expressions is common in practice; in particular, such expressions are part of the IEEE standards for temporal logics PSL [10, 11] and SVA [7]. Furthermore, using the fusion operator of PSL, that concatenates two regular expressions with one letter overlap, the two positive schematic examples above can be merged into the single expression $\mathtt{true}^* \circ (p{\wedge}q) \circ \mathtt{true}^* \circ (r) \circ \mathtt{true}^* \circ (p{\wedge}q)$.

It is well known that, for every LTL formula φ, $[\![\varphi]\!]^{\mathrm{fin}}$ and $[\![\neg\varphi]\!]^{\mathrm{fin}}$ are regular languages. A classic result in [24] states that every regular language is a union of finitely many languages definable by union-free regular expressions. The union-free regular expressions in question may, however, have unbounded star height. Hence, this observation alone does not yield finite unique characterizations consisting of schematic examples.

The results in this section are restricted to LTL interpreted over finite words. For instance, $\mathtt{F}p$ and $\mathtt{F}(p \wedge \neg\hat{\mathtt{F}}p)$ are non-equivalent in general, but cannot be distinguished by finite words, and therefore also not by schematic examples. In this sense, our results that use transfinite examples still offer benefits beyond those of schematic examples. Even over $\mathcal{W}^{\mathsf{fin}}$, we believe that the full language of LTL does not admit finite characterizations by schematic examples, but this remains to be proved.

8 Another notion of "good" examples: characteristic samples

The main focus of this paper is on the existence of *unique characterizations* for fragments of LTL. In this section, we briefly compare this notion with another notion of a "good" set of examples, namely *characteristic samples*. Let a *concept class* be a triple $\mathbb{C} = (C, \mathcal{X}, [\![\cdot]\!])$, where C is a set (the *concepts*), $\mathcal{X}$ is a set (the *examples*) and where, for each $c \in C$, $[\![c]\!] \subseteq \mathcal{X}$.

▶ **Definition 8.1** (Learner and teacher)**.** *Fix a concept class* $\mathbb{C} = (C, \mathcal{X}, [\![\cdot]\!])$.

- *A* learner *for* $\mathbb{C}$ *is a function* $\boldsymbol{L} : \wp_{\mathit{finite}}(\mathcal{X} \times \{0,1\}) \to \Psi$ *with the property that* $\boldsymbol{L}(S)$ *fits* S*, assuming* S *fits at least one element of* $\mathbb{C}$.[3]
- *A* teacher *for* $\mathbb{C}$ *is a function* $\boldsymbol{T} : \Psi \to \wp_{\mathit{finite}}(\mathcal{X} \times \{0,1\})$ *such that* $\boldsymbol{T}(\psi)$ *fits* ψ*.*

Note that learners and teachers need not be computable.

▶ **Definition 8.2** (Characteristic Samples)**.**

- *A sample* S *is a* characteristic sample *for* ψ *and a learner* $\boldsymbol{L}$ *if* S *fits* ψ *and for every sample* $S' \supseteq S$ *that fits* ψ *we have* $[\![\boldsymbol{L}(S')]\!] = [\![\psi]\!]$*. The intuition is that additional information consistent with* ψ *beyond* S *will not cause the learner to change its mind about the correct concept.*

[3] If S fits no element of $\mathbb{C}$ then there is no requirement on $\boldsymbol{L}(S)$.

- *A concept class* $\mathbb{C}$ has characteristic samples for a learner $\boldsymbol{L}$ *if there exists a teacher* $\boldsymbol{T}$ *such that for every* ψ *in* $\mathbb{C}$*,* $\boldsymbol{T}(\psi)$ *is a characteristic sample for* ψ *and* $\boldsymbol{L}$*. A concept class* $\mathbb{C}$ has characteristic samples *if it has characteristic samples for some learner.*

The following is a known property of characteristic samples.

▶ **Lemma 8.3.** *Let* $\mathbb{C} = (\Psi, \mathcal{X}, [\![\cdot]\!])$ *be a concept class. Assume* $\mathbb{C}$ *has characteristic samples with respect to learner* $\boldsymbol{L}$ *and teacher* $\boldsymbol{T}$*. Let* $\psi, \psi' \in \Psi$*. If* $[\![\psi]\!] \neq [\![\psi']\!]$ *then there exists* $(x, b) \in \boldsymbol{T}(\psi) \cup \boldsymbol{T}(\psi')$ *such that* $x \in [\![\psi]\!] \oplus [\![\psi']\!]$ *where* $\oplus$ *denotes the symmetric difference.*

Proof. Assume this is not the case. Let S be a sample subsuming $\boldsymbol{T}(\psi) \cup \boldsymbol{T}(\psi')$. Let $\psi'' = \boldsymbol{L}(S)$. Since S fits ψ and subsumes $\boldsymbol{T}(\psi)$, by the definition of characteristic samples, ψ'' should satisfy $[\![\psi'']\!] = [\![\psi]\!]$. The same reasoning entails that $[\![\psi'']\!] = [\![\psi']\!]$, contradicting $[\![\psi]\!] \neq [\![\psi']\!]$. ◀

The notion of unique characterization is stronger than that of characteristic samples, as formally stated in the following theorem.

▶ **Theorem 8.4.** *If a concept class* $\mathbb{C}$ *admits a unique characterization, then it has characteristic samples.*

It follows that all fragments for which we have seen positive results regarding unique characterizations also have characteristic samples. The converse is not true, since as the following theorem shows the concept class consisting of all LTL formulas has characteristic samples, while we have provided several negative results regarding unique characterization for fragments of LTL.

▶ **Theorem 8.5.** *Let* AP *be any finite set of atomic propositions.* $\mathsf{LTL}[\mathsf{AP}]$ *admits characteristic samples over* $\mathcal{W}^{\mathsf{fin}}$*.*

The theorem can be proved using Gold's enumeration technique [16].

This stands in sharp contrast with the results in Section 4, which showed that finite unique characterizations over $\mathcal{W}^{\mathsf{fin}}$ rarely exist.

In a sense, as evident from the respective definitions and proofs, unique characterization can be viewed as a strengthening of the notion of characteristic samples, requiring the property to hold with respect to any learner rather than a particular one.

Another aspect in which the two notions can be compared is the size of the example set. We measure the size of a set as the sum of the lengths of the examples it contains. Gold's method, used in the proof of Theorem 8.5, yields characteristic samples of exponential cardinality (that is, the number of examples is exponential, regardless of their size). The result of Barzdin and Freivalds [3] shows that characteristic samples of polynomial cardinality are possible. This result, however, also does not take into account the size of the examples.

Regarding the size of the examples themselves, we can obtain a doubly exponential upper bound by translating the formula into an automaton and taking a characteristic sample for it. An LTL formula of size n can be translated into a Büchi automaton B_ψ of size $2^{O(n)}$. The language L of a Büchi automaton can be represented as the following language of finite words $(L)_\$ = \{u\$v \mid u(v)^\omega \in L\}$ since two ω-regular languages are equivalent iff they agree on the set of ultimately periodic. Calbrix et al. [5] showed that $(L)_\$$ is a regular language of finite words. Kuperberg et al. [20] showed that a non-deterministic Büchi automaton with n states can be converted to a DFA with at most $2^n + 3^{n^2}$. Thus, given an LTL formula ψ of size n, we can obtain a DFA for $(L(B_\psi))_\$$ whose size is doubly exponential in n. Since a characteristic sample for a DFA has size polynomial in the number of its states, the overall doubly exponential upper bound follows. We complement this by a matching lower bound.

▶ **Proposition 8.6.** *There exists a family of languages* $\{L_n\}_{n \in \mathbb{N}}$ *that can be represented by LTL formulas of size linear in* n *whereas the DFA for* $(L_n)_\$$ *requires at least* 2^{2^n} *states.*

This opens the question whether working with some other representation for ω-regular languages may yield a smaller size automaton. We note that the answer is negative for any deterministic ω-automaton (resp. family of DFAs used to represent ω-regular languages [22, 2, 12]), since the same proof as above goes through (resp. for the leading automaton).

Thus, the doubly-exponential bound matches the complexity of obtaining a characteristic sample for LTL via a corresponding deterministic automaton. It remains possible, however, that more efficient constructions of characteristic samples for LTL formulas exist that do not proceed through such automata representations.

9 Conclusion

We studied when LTL-formulas can be uniquely characterized by finite sets of labeled examples. We gave a complete classification, over finite words, of the operator fragments that admit such finite unique characterizations, showing that this property is quite limited in this setting. We then showed that moving to richer example domains changes the picture substantially: while ω-words already help for some fragments, transfinite words enable finite unique characterizations for large monotone fragments of LTL. Finally, returning to the finite setting, we introduced schematic examples as compact representations of families of finite words, enabling unique characterization results for large monotone fragments of LTL.

Together, these results provide a systematic account of how the descriptive power of examples for LTL-formulas depends on the type of examples that is allowed. They also suggest that schematic examples are a promising middle ground between concrete traces and more abstract symbolic descriptions: they remain close to the kinds of finite behaviors practitioners use, while being quite expressive in terms of describing the behavior of an LTL-formula.

Several directions remain open. On the technical side, it would be natural to extend the present analysis to larger fragments involving both $\mathtt{F}$ and $\mathtt{G}$. Furthermore, restricting the set of allowed operators is a natural first step, but one could also systematically study other kinds of syntactic fragments (such as obtained by forbidding specific operator nesting patterns, as is done in some of the results in [13]). On the conceptual side, it would be interesting to revisit related notions such as characteristic samples, active learning, as well as interfaces for debugging or explanation, through the lens of richer example formats, and to better understand when compact, human-interpretable examples can serve as faithful descriptions of temporal specifications.

References

1 Glenn Ammons, David Mandelin, Rastislav Bodík, and James R. Larus. Debugging temporal specifications with concept analysis. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 182–195. ACM, 2003. doi:10.1145/781131.781152.

2 Dana Angluin and Dana Fisman. Learning regular omega languages. *Theor. Comput. Sci.*, 650:57–72, 2016. doi:10.1016/J.TCS.2016.07.031.

3 J. M. Barzdin and R. V. Freivalds. On the prediction of total recursive functions. *Doklady Akademii Nauk SSSR*, 206(3):521–524, 1972. (In Russian).

4 Benjamin Bordais, Daniel Neider, and Rajarshi Roy. The complexity of learning LTL, CTL and ATL formulas. In Olaf Beyersdorff, Michal Pilipczuk, Elaine Pimentel, and Kim Thang Nguyen, editors, *42nd International Symposium on Theoretical Aspects of Computer Science, STACS 2025, Jena, Germany, March 4-7, 2025*, LIPIcs, pages 19:1–19:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.STACS.2025.19.

5 Hugues Calbrix, Maurice Nivat, and Andreas Podelski. Ultimately periodic words of rational w-languages. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings*, volume 802 of *Lecture Notes in Computer Science*, pages 554–566. Springer, 1993. doi:10.1007/3-540-58027-1_27.

6 Andrés Camacho and Sheila A. McIlraith. Learning interpretable models expressed in linear temporal logic. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2019.

7 Eduard Cerny, Surrendra Dudani, John Havlicek, and Dmitry Korchemny. *SVA: The Power of Assertions in SystemVerilog*. Springer, 2014.

8 Stéphane Demri and David Nowak. Reasoning about transfinite sequences. *International Journal of Foundations of Computer Science*, 18(01):87–112, 2007. doi:10.1142/S0129054107004589.

9 Stephane Demri and Alexander Rabinovich. The complexity of linear-time temporal logic over the class of ordinals. *Logical Methods in Computer Science*, Volume 6, Issue 4, December 2010. doi:10.2168/LMCS-6(4:9)2010.

10 Cindy Eisner and Dana Fisman. *A Practical Introduction to PSL*. Series on Integrated Circuits and Systems. Springer, 2006. doi:10.1007/978-0-387-36123-9.

11 Cindy Eisner and Dana Fisman. Functional specification of hardware via temporal logic. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 795–829. Springer, 2018. doi:10.1007/978-3-319-10575-8_24.

12 Dana Fisman, Emmanuel Goldberg, and Oded Zimerman. A robust measure on fdfas following duo-normalized acceptance. In Rastislav Královic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, Bratislava, Slovakia, August 26-30, 2024*, volume 306 of *LIPIcs*, pages 53:1–53:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.MFCS.2024.53.

13 Marie Fortin, Boris Konev, Vladislav Ryzhikov, Yury Savateev, Frank Wolter, and Michael Zakharyaschev. Unique characterisability and learnability of temporal instance queries. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, pages 163–173, 2022. doi:10.24963/kr.2022/17.

14 Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Frontiers and exact learning of ELI queries under DL-Lite ontologies. In *IJCAI*, pages 2627–2633. ijcai.org, 2022. doi:10.24963/IJCAI.2022/364.

15 Ivan Gavran, Eva Darulova, and Rupak Majumdar. Interactive synthesis of temporal specifications from examples and natural language. *Proceedings of the ACM on Programming Languages*, 4(OOPSLA):201:1–201:26, 2020. doi:10.1145/3428269.

16 E.M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967. doi:10.1016/S0019-9958(67)91165-5.

17 Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995. doi:10.1006/jcss.1995.1003.

18 Christopher Hahn, Frederik Schmitt, Jens U. Kreber, Markus Norman Rabe, and Bernd Finkbeiner. Teaching temporal logics to neural networks. In *International Conference on Learning Representations*, 2021. URL: https://openreview.net/forum?id=dOcQK-f4byz.

19 Jean Christoph Jung, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Unique characterisability and learnability of temporal queries mediated by an ontology. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*, pages 487–497, 2024. doi:10.24963/kr.2024/46.

20 Denis Kuperberg, Laureline Pinault, and Damien Pous. Coinductive algorithms for Büchi automata. In Piotrek Hofman and Michal Skrzypczak, editors, *Developments in Language Theory - 23rd International Conference, DLT 2019, Warsaw, Poland, August 5-9, 2019, Proceedings*, volume 11647 of *Lecture Notes in Computer Science*, pages 206–220. Springer, 2019. doi:10.1007/978-3-030-24886-4_15.

21 Runming Li, Keerthana Gurushankar, Marijn J. H. Heule, and Kristin Y. Rozier. What's in a name? Linear Temporal Logic literally represents time lines. In *Proceedings of the IEEE Working Conference on Software Visualization (VISSOFT)*, pages 73–83, 2023. doi:10.1109/VISSOFT60811.2023.00018.

22 Oded Maler and Ludwig Staiger. On syntactic congruences for omega-languages. In Patrice Enjalbert, Alain Finkel, and Klaus W. Wagner, editors, *STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science, Würzburg, Germany, February 25-27, 1993, Proceedings*, volume 665 of *Lecture Notes in Computer Science*, pages 586–594. Springer, 1993. doi:10.1007/3-540-56503-5_58.

23 Corto Mascle, Nathanaël Fijalkow, and Guillaume Lagarde. Learning temporal formulas from examples is hard. *CoRR*, abs/2312.16336, 2023. doi:10.48550/arXiv.2312.16336.

24 Benedek Nagy. A normal form for regular expressions. In C. S. Calude, E. Calude, and M. J. Dinnen, editors, *Supplemental Papers for DLT'04*, Auckland, New Zealand, 2004. CDMTCS.

25 Daniel Neider and Ivan Gavran. Learning linear temporal properties. In *Formal Methods in Computer-Aided Design (FMCAD)*, pages 1–10. IEEE, 2018. doi:10.23919/FMCAD.2018.8603016.

26 Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57. IEEE, 1977. doi:10.1109/SFCS.1977.32.

27 Ritam Raha, Rajarshi Roy, Nathanaël Fijalkow, and Daniel Neider. A scalable anytime algorithm for learning fragments of linear temporal logic. *Formal Methods Syst. Des.*, 68(1):2, 2026. doi:10.1007/S10703-025-00489-Y.

28 Viktor Schuppan and Armin Biere. Shortest counterexamples for symbolic model checking of LTL with past. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 3440 of *Lecture Notes in Computer Science*, pages 493–509. Springer, 2005. doi:10.1007/978-3-540-31980-1_32.

29 Balder ten Cate and Víctor Dalmau. Conjunctive queries: Unique characterizations and exact learnability. *ACM Trans. Database Syst.*, 47(4), November 2022. doi:10.1145/3559756.

30 Balder ten Cate, Dana Fisman, Roi Ohayon, and Patrik Sestic. Characterizing LTL formulas by examples. *CoRR*, abs/2604.22097, 2026. doi:10.48550/arXiv.2604.22097.

31 Moshe Y. Vardi. An automata-theoretic approach to linear temporal logic. In Faron Moller and Graham M. Birtwistle, editors, *Logics for Concurrency - Structure versus Automata (8th Banff Higher Order Workshop, Banff, Canada, August 27 - September 3, 1995, Proceedings)*, Lecture Notes in Computer Science, pages 238–266. Springer, 1995. doi:10.1007/3-540-60915-6_6.

32 Zili Wang, Laura P. Gamboa Guzman, and Kristin Y. Rozier. WEST: Interactive validation of mission-time linear temporal logic (MLTL). *Science of Computer Programming*, 248:103365, 2026. doi:10.1016/j.scico.2025.103365.

33 Zhe Xu, Yuxin Chen, and Ufuk Topcu. Adaptive teaching of temporal logic formulas to preference-based learners. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 5061–5068. AAAI Press, 2021. doi:10.1609/AAAI.V35I6.16640.

34 Changjian Zhang, Parv Kapoor, Ian Dardik, Leyi Cui, Rômulo Meira-Góes, David Garlan, and Eunsuk Kang. Constrained ltl specification learning from examples. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, ICSE '25, pages 629–641. IEEE Press, 2025. doi:10.1109/ICSE55347.2025.00160.