On Finite Domains in First-Order Linear Temporal Logic*

Julien Brunel[1], David Chemouil[1], and Denis Kuperberg[1,2]

[1] Onera/DTIM, Toulouse
[2] IRIT, University of Toulouse

Abstract. We consider First-Order Linear Temporal Logic (FO-LTL) over linear time $\mathbb{N}$. We focus on finding models with finite first-order domains for FO-LTL formulas. More precisely, we investigate the complexity of the following problem: given a formula φ and an integer n, is there a model of φ with domain of cardinality at most n? We show that depending on the logic considered (FO or FO-LTL) and on the precise encoding of the problem, the problem is either NP-complete, NEXPTIME-complete, PSPACE-complete or NEXPSPACE-complete. In a second part, we exhibit cases where the Finite Model Property can be lifted from fragments of FO to their FO-LTL extension.

1 Introduction

First-Order Logic (FO) has proven to be useful in numerous applications in computer science such as formal specification, databases, ontology languages, etc. It is particularly well-suited to reason about objects of a domain, their relations and the properties they satisfy. However, since "full" FO is undecidable, the formal *verification* of properties implies a relaxation of the problem *e.g.* considering less expressive fragments. Thus, one can restrict the specification language (*e.g.* Prolog) or impose some form of interaction for verification (*e.g.* theorem provers, proof assistants).

Another form of trade-off is to keep the whole logic and full automation but to rely on a sound but incomplete decision procedure. For instance, the Analyzer[3] for the Alloy [Jac06] language (based upon relational first-order logic) implements a *bounded-satisfiability* decision procedure. That is, the tool is a *"finite-model finder":* it first bounds the number of objects in the domain and then runs a classical propositional SAT procedure. Thanks to the performance of modern SAT engines, this approach has showed to be very efficient in practice to find counterexamples quickly when assessing specifications. This is one of the reasons for the success of Alloy in the formal methods community.

However, in most software and systems specifications, one needs to represent the evolution of modeled entities along time. In Alloy, a common way to do so is

* Research funded by ANR/DGA project Cx (ref. ANR-13-ASTR-0006); and by *fondation STAE* project BRIefcaSE.

[3] Available at `http://alloy.mit.edu/alloy`.

to model time by adding a specific set of instants, by giving axioms describing its structure (as traces for instance) and by adding an extra time parameter to every dynamic predicate. This is tedious and cumbersome, if not error-prone. Furthermore, the number of instants gets bounded all the same as for other domain objects: this is a pity as one could rely on a temporal logic such as LTL and then enjoy complete decision procedures.

These remarks led us to study the combination of FO and LTL, in particular to draw questions about the relation between the satisfiability of a FO-LTL formula and the fact that the first-order part of the model is finite. In the literature, the logic FO-LTL has drawn a lot of interest, for decidability questions as well as in database theory [AHdB96]. For instance [HWZ00, HKK$^+$03] study decidable fragments, while [Kam68, Mer92] give incompleteness results.

2 The NSAT problem

The first question we address here is that of the complexity of satisfiability for FO-LTL when the FO part of the model is bounded by an integer N which is part of the input (we call this problem "NSAT"). Interestingly, it seems that this problem has not been clearly addressed in the literature, even for the pure FO case, although we believe for this case it can be considered as folklore. We consider NSAT for FO and FO-LTL depending on whether the quantifier rank (*i.e.* the maximum number of nested quantifiers) of formulas is bounded and whether N is given in unary or binary encoding. For pure FO, we show that NSAT is NP-complete for bounded rank and unary N, and NEXPTIME-complete otherwise. These results are reproduced here for compleness, although they might be already known by the community. For FO-LTL, which has been less studied from the point of view of NSAT, we show that this division goes the same except that NSAT is PSPACE-complete in the first case and EXPSPACE-complete otherwise (recall that satisfiability for LTL alone is PSPACE-complete [SC85, LP85]).

The results are summed up in the following theorem:

Theorem 1. *We consider NSAT for three parameters: logic, encoding, bound on* $\mathrm{rk}(\varphi)$ (ranked)*. The corresponding complexities are given in the following table:*

	N unary	*N binary*
FO ranked	*NP-complete*	*NEXPTIME-complete*
FO	*NEXPTIME-complete (even N = 2)*	*NEXPTIME-complete*
FO-LTL ranked	*PSPACE-complete*	*EXPSPACE-complete*
FO-LTL	*EXPSPACE-complete (even N = 2)*	*EXPSPACE-complete*

3 Finite Model Property

3.1 Lifting from FO to FO-LTL

Secondly, since we are only interested in finite models of FO-LTL formulas, it is natural to study which fragments of FO-LTL enjoy the *finite model property* (FMP). Recall that a formula has the FMP if the existence of a model

implies the existence of a *finite* model. Many fragments of FO have been shown to enjoy the FMP in the past decades [BGG97, ARS07]. We show that any fragment of FO enjoying the FMP and verifying a certain condition called refinement (resp. plus-refinement) can be "lifted" as a fragment of FO-LTL using also **X** (resp. **X** and **F**) and still enjoying the FMP (provided the removal of the temporal operators leads back to the original FO fragment). These conditions states that the fragment does not impose strong constraints on the number or arity of predicates and functions. They are met by most fragments studied in practice, although not all of them. Here we consider only formulas where negations have been pushed to the leaves, so **G** cannot be obtained by negation of **F** in the lifted fragments.

For instance in the following list of fragments with FMP, only the Grädel fragment does not verify the plus-refinement condition, because it requires that there is at most one unary function in the signature.

Example 1. [BGG97, ARS07] The following fragments of FO, named following the notation of [BGG97], have the FMP:

- $[\exists^*\forall^*, all]_=$ (Ramsey 1930) the class of all sentences with quantifier prefix $\exists^*\forall^*$ over arbitrary relational vocabulary with equality.
- $[\exists^*\forall\exists^*, all]_=$ (Ackermann 1928) the class of all sentences with quantifier prefix $\exists^*\forall\exists^*$ over arbitrary relational vocabulary with equality.
- $[\exists^*, all, all]_=$ (Gurevich 1976) the class of all sentences with quantifier prefix $\exists^*$ over arbitrary vocabulary with equality.
- $[\exists^*\forall, all, (1)]_=$ (Grädel 1996) the class of all sentences with quantifier prefix $\exists^*\forall$ over vocabulary that contain one unary function and arbitrary predicate symbols with equality.
- FO_2 (Mortimer 1975) the class of all sentences of relational vocabulary that contain two variables and equality.

3.2 Axioms of infinity

We finally show that with temporal operators **G** or **U**, the FMP is lost, even with strong constraints on the way temporal operators interact with first-order quantifiers.

The following FO-LTL sentence shows that one existential variable is enough to force an infinite first-order structure:

$$\mathbf{G}(\exists x.P(x) \wedge \mathbf{X}(\mathbf{G}\,\neg P(x)))).$$

The previous example works because the $\exists$ quantifier is nested inside a **G** operator, and is therefore "called" infinitely many times. The following sentence shows that the same result can be achieved without nesting any first-order quantifier inside temporal operators (c is a constant in the signature):

$$\forall x \exists y.P(c) \wedge \mathbf{G}(P(x) \Rightarrow \mathbf{X}(P(y) \wedge \mathbf{G}\,\neg P(x))).$$

Finally, the two previous examples used the **G** operator that imposes a constraint on infinitely many instants, thereby generating infinitely many first-order

elements. It is possible to achieve the same result using **U** which only imposes a constraint on a finite (but unbounded) number of time instants:

$$\forall x \exists y. P(c) \wedge ((P(x) \wedge P(y)) \,\mathbf{U}\, (\neg P(x) \wedge P(y))).$$

Disclaimer: This work has been submitted to MFCS 2015.

References

[AHdB96] Serge Abiteboul, Laurent Herr, and Jan Van den Bussche. Temporal versus first-order logic to query temporal databases. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, 1996, Montreal, Canada*, pages 49–57, 1996.

[ARS07] Aharon Abadi, Alexander Moshe Rabinovich, and Mooly Sagiv. Decidable fragments of many-sorted logic. In *14th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2007)*, pages 17–31, 2007.

[BGG97] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[HKK+03] Ian M. Hodkinson, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *10th International Symposium on Temporal Representation and Reasoning, / 4th International Conference on Temporal Logic (TIME-ICTL 2003)*, pages 91–98, 2003.

[HWZ00] Ian Hodkinson, Frank Wolter, and Michael Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106(1–3):85 – 134, 2000.

[Jac06] Daniel Jackson. *Software Abstractions - Logic, Language, and Analysis*. MIT Press, 2006.

[Kam68] Hans W. Kamp. *Tense Logic and the Theory of Linear Order*. Phd thesis, University of Warsaw, 1968.

[LP85] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proceedings of the 12th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 97–107. ACM, 1985.

[Mer92] Stephan Merz. Decidability and incompleteness results for first-order temporal logics of linear time. *Journal of Applied Non-Classical Logics*, 2(2):139–156, 1992.

[SC85] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.