Checking Satisfiability of CLTL without Automata

Marcello M. Bersani, Achille Frigeri

Politecnico di Milano bersani@elet.polimi.it,achille.frigeri@polimi.it

1 Introduction

Finite-state system verification has attained great successes, both using automata-based and logic-based techniques. Examples of the former are the so-called explicit-state model checkers [1] and symbolic model checkers [2]. However, some of the best results have been obtained by logic-based techniques, such as Bounded Model Checking (BMC) [3], a fully automated (although potentially incomplete) procedure. In BMC, a finite-state machine A (typically, a version of Büchi Automata) and a desired property P expressed in Propositional Linear Temporal Logic (PLTL) are translated into a Boolean formula ϕ to be fed to a SAT solver. The translation is made finite by bounding the number of time instants. However, infinite behaviors, which are crucial in proving, e.g., liveness properties, are also considered by using the well-known property that a Büchi Automaton accepts an infinite behavior if, and only if, it accepts an infinite periodic behavior. Hence, chosen a bound $k > 0$, a Boolean formula ϕ_k is built, such that ϕ_k is satisfiable if and only if there exists an infinite periodic behavior of the form $\alpha\beta^\omega$, with $|\alpha\beta| \leq k$, that is compatible with system A while violating property P. This procedure allows counterexample detection, but it is not complete, since the violations of property P requiring "longer" behaviors, i.e., of the form $\alpha\beta^\omega$ with $|\alpha\beta| > k$, are not detected. However, in many practical cases it is possible to find bounds large enough for representing counterexamples, but small enough so that the SAT solver can actually find them in a reasonable time.

Clearly, the BMC procedure can be used to check satisfiability of a PLTL formula, without considering a finite state system A. This has practical applications, since a PLTL formula can represent both the system and the property to be checked (see, e.g., [4], where the translation into Boolean formulae is made more specific for dealing with satisfiability checking and metric temporal operators). We call this case *Bounded Satisfiability Checking* (BSC), which consists in solving a so-called Bounded Satisfiability Problem: Given a PLTL formula P, and chosen a bound $k > 0$, define a Boolean formula ϕ_k such that ϕ_k is satisfiable if, and only if, there exists an infinite periodic behavior of the form $\alpha\beta^\omega$, with $|\alpha\beta| \leq k$, that satisfies P.

The introduction of many extensions of temporal logic proposed in order to express property of *infinite*-state systems, has lead to the study of CLTL($\mathcal{D}$), a general framework extending the future-fragment of PLTL by allowing arithmetic constraints belonging to a generic constraint system $\mathcal{D}$. The resulting logics are expressive and well-suited to define infinite-state systems and their properties, but, even for the bounded case, their satisfiability is typically undecidable [5], since they can simulate general two-counter machines when $\mathcal{D}$ is powerful enough (e.g., Difference Logic). However, there are some decidability results, which allow in principle for some kind of automatic verification. Most notably, satisfiability of CLTL($\mathcal{D}$) is decidable (in PSPACE) when $\mathcal{D}$ is the class of Integer Periodic Constraints (IPC*) [6], or when it is the structure $(D, <, =)$ with $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ [7]. In these cases, decidability is shown by using an automata-based approach similar to the standard case for LTL, by reducing satisfiability checking to emptiness verification of Büchi automata. Given a CLTL($\mathcal{D}$) formula ϕ, with $\mathcal{D}$ as in the above cases, it is in fact possible to define an automaton $\mathcal{A}_\phi$ such that ϕ is satisfiable if, and only if, the language recognized by $\mathcal{A}_\phi$ is not empty. These results, although of great theoretical interest, are not well suited for a direct implementation, since the involved constructions are very inefficient.

We extended [8] the above results to a more general logic, called CLTLB($\mathcal{D}$), which is an extension of PLTLB (PLTL with Both future and past operators) with arithmetic constraints in constraint system $\mathcal{D}$, and consider a procedure for satisfiability verification that does not rely on automata constructions. This procedure is implemented in the $\mathbb{Z}$ot toolkit, verified by standard SMT-solvers, such as z3 [9].

The idea of the procedure is to verify satisfiability by checking a finite number of k-satisfiability problems. Informally, k-satisfiability amounts to looking for ultimately periodic *symbolic* models of the form $\alpha\beta^\omega$, i.e., such that prefix $\alpha\beta$ of length k admits a bounded arithmetic model (up to instant k). Although the k-bounded problem is defined with respect to a bounded arithmetical model, it provides a finite representation of infinite symbolic models by means of ultimately periodic words. When CLTLB($\mathcal{D}$) has the property that its ultimately periodic symbolic models, of the form $\alpha\beta^\omega$, always admit an arithmetic model, then the k-satisfiability problem can be reduced to satisfiability of QF-EU$\mathcal{D}$ (the theory of quantifier-free equality and uninterpreted functions combined with $\mathcal{D}$). In this case, k-satisfiability is equivalent to satisfiability over infinite models.

Symmetrically to standard LTL, where bounded model-checking and SAT-solvers can be used as an alternative to automata-theoretic approaches to model-checking, reducing satisfiability to k-satisfiability allows SMT-solvers to be used in solving satisfiability for CLTLB($\mathcal{D}$) formulae, instead of checking emptiness of a Büchi automaton. Moreover, when the length of all prefixes $\alpha\beta$ to be tested is bounded by some finite K, then the number of bounded problems to be solved is also bounded. Therefore, we also proved that k-satisfiability is *complete* with respect to the satisfiability problem, i.e., by checking at most K bounded problems satisfiability of CLTLB($\mathcal{D}$) formulae can always be answered.

2 Bounded Satisfiability Problem

The k-satisfiability problem for CLTLB($\mathcal{D}$) formulae is defined in terms of the existence of a so-called k-bounded arithmetical model σ_k, which provides a finite representation of infinite symbolic models by means of ultimately periodic words. This allows to prove that k-satisfiability is still representative of the satisfiability problem. In fact, for some constraint systems, a bounded solution can be used to build the infinite model σ for the formula from the k-bounded one σ_k and from its symbolic model. We showed that a formula ϕ is satisfiable if, and only if, it is k-satisfiable and its bounded solution σ_k can be used to derive its infinite model σ. In case of negative answer to a k-bounded instance, we can not immediately entail the unsatisfiability of the formula. However, we proved that for every formula ϕ there exists an upper bound K, which can effectively be determined, such that if ϕ is not k-satisfiable for all k in $[1, K]$, then ϕ is unsatisfiable.

A bounded symbolic model is, informally, a finite representation of infinite CLTLB($\mathcal{D}$) models over the alphabet of symbolic valuations $SV(\phi)$. We restrict the analysis to ultimately periodic symbolic models, i.e., of the form $\rho = \alpha(\beta)^\omega$. The Bounded Satisfiability Problem (BSP) is defined with respect to a k-bounded model σ_k (i.e., an assignment for variable in the first k-instants), a finite sequence ρ' (with $|\rho'| = k+1$) of symbolic valuations and a k-bounded satisfaction relation $\models_k$ defined as follows:

$$\sigma_k, 0 \models_k \rho' \text{ iff } \sigma_k, i \models \rho'(i) \text{ for all } 0 \leq i \leq k.$$

The *k-satisfiability problem* of formula ϕ is defined as follows:

Input A CLTLB($\mathcal{D}$) formula ϕ, a constant $k \in \mathbb{N}$

Problem Is there an ultimately periodic sequence of symbolic valuations $\rho = \alpha(\beta)^\omega$ (with $|\alpha\beta| = k+1$), such that $\rho, 0 \models \phi$ and which admits a k-bounded model σ_k such that $\sigma_k \models_k \rho'$, with $\rho' = \alpha\beta$?

Since the length k is fixed, the procedure for determining the satisfiability of CLTLB($\mathcal{D}$) formulae over bounded models is not complete: even if there is no accepting run of automaton $\mathcal{A}_\phi$ when ρ' as above has length k, there may be accepting runs for a larger ρ'.

Definition 1. *Given a CLTLB($\mathcal{D}$) formula ϕ, its* completeness threshold K_ϕ, *if it exists, is the smallest number such that ϕ is satisfiable if and only if ϕ is K_ϕ-satisfiable.*

Theorem 1. *Let ϕ be a CLTLB($\mathcal{D}$) formula. If $\mathcal{D}$ is $(D, <, =)$, then the completeness threshold exists and is less then $|SV(\phi)| \cdot 2^{|\phi|}$. If $\mathcal{D}$ is IPC*, then the completeness threshold exists and is less then $4|V|^2|\lambda|^2|SV(\phi)| \cdot 2^{|\phi|}$, where λ is an effectivly constant depending on the depth of ϕ.*

3 Encoding for BSP without Automata

We proved that the BSP for a CLTLB($\mathcal{D}$) formula can be reduced to the satisfiability of a quantifier-free formula in the theory EUF $\cup\, \mathcal{D}$ (QF-EU$\mathcal{D}$), where EUF is the theory of Equality and Uninterpreted Functions, provided that $\mathcal{D}$ includes a copy of $\mathbb{N}$ with the successor relation and that EUF $\cup\, \mathcal{D}$ is consistent. The last condition is easily verified in the case of the union of two consistent, disjoint, stably infinite theories (as is the case for EUF and arithmetic). In [10] a similar approach is described for the case of Integer Difference Logic (DL) constraints. It is worth noting that standard LTL can be encoded by a formula in QF-EU$\mathcal{D}$ with $\mathcal{D} = (\mathbb{N}, <)$. In this case, the encoding is more succinct than the Boolean one proposed in [11].

We denote the encoding of the BSP for ϕ with bound k by $|\phi|_k$. We proved the main equivalence result which draws the connection between such encoding and the k-satisfiability problem.

Theorem 2. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation, ϕ is k-satisfiable with respect to $k \in \mathbb{N}$ if, and only if, $|\phi|_k$ is satisfiable.*

Proposition 1. *Let $\phi \in$ CLTLB($\mathcal{D}$) with $\mathbb{N}$ definable in $\mathcal{D}$ together with the successor relation, ϕ is k-satisfiable with respect to $k \in \mathbb{N}$ if, and only if, ϕ has an ultimately periodic model $\alpha\beta^\omega$ with $|\alpha\beta| = k + 1$.*

References

1. Holzmann, G.: The model checker SPIN. IEEE Transactions on Software Engineering **23**(5) (may 1997) 279 –295
2. Clarke, E., McMillan, K., Campos, S., Hartonas-Garmhausen, V.: Symbolic model checking. In: Computer Aided Verification. Volume 1102 of Lecture Notes in Computer Science. (1996) 419–422
3. Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic model checking without BDDs. In: Tools and Algorithms for the Construction and Analysis of Systems. Volume 1579 of Lecture Notes in Computer Science. (1999) 193–207
4. Pradella, M., Morzenti, A., San Pietro, P.: Bounded satisfiability checking of metric temporal logic specifications. ACM Transactions on Software Engineering and Methodology (TOSEM) (2012) To appear.
5. Demri, S., Gascon, R.: The effects of bounding syntactic resources on Presburger LTL. Technical Report LSV-06-5, LSV (2006)
6. Demri, S., Gascon, R.: The effects of bounding syntactic resources on Presburger LTL. In: International Syposium on Temporal Representation and Reasoning (TIME), IEEE Computer Society (2007) 94–104
7. Demri, S., D'Souza, D.: An automata-theoretic approach to constraint LTL. Information and Computation **205**(3) (2007) 380–415
8. Bersani, M.M., Frigeri, A., Morzenti, A., Pradella, M., Rossi, M., San Pietro, P.: Constraint LTL Satisfiability Checking without Automata. ACM Transactions on Computational Logic (submitted)
9. Microsoft Research: Z3: An efficient SMT solver. http://research.microsoft.com/en-us/um/redmond/projects/z3/ (2009)
10. Bersani, M.M., Cavallaro, L., Frigeri, A., Pradella, M., Rossi, M.: SMT-based verification of LTL specification with integer constraints and its application to runtime checking of service substitutability. In: IEEE International Conference on Software Engineering and Formal Methods. (2010) 244–254
11. Biere, A., Heljanko, K., Junttila, T.A., Latvala, T., Schuppan, V.: Linear encodings of bounded LTL model checking. Logical Methods in Computer Science **2**(5) (2006)