

# Towards a Notion of Unsatisfiable and Unrealizable Cores for LTL

Viktor Schuppan

## Contribution

- Finding unsatisfiable cores in temporal logic

## Methods

Find unsat cores from

- Structure of Syntax tree
- Converting formula into normal form
- Pull out of proofs that show the formula is unsatisfiable

## Previous work and Problem

Current implementations will only give that

$$\phi = (\mathcal{G}\psi) \wedge (\mathcal{F}\psi')$$

is the unsat core. Schuppan looks inside of  $\mathcal{G}$  and  $\mathcal{F}$  and all of the temporal operators.

This work uses finer grained methods to say that "this little piece" is the problem.

## Computing minimal unsatisfiable core for LTL over finite traces [5]

### Paper Layout

- Introduction
- Preliminaries
- Computing  $LTL_f$  minimal unsatisfiable core
- Experimental evaluation
- Concluding remarks

## Introduction

- $LTL_f$  is LTL but over finite traces
- discusses how deeply studied  $LTL_f$  has been
  - fundamental techniques for  $LTL_f$  reasoning, e.g. satisfiability checking [citations], the translation to automata [citations] and synthesis [citations] have been investigated in depth.
- If the checking result turns out to be unsatisfiable, the specification is meaningless and has to be amended. Such procedure is called specification debugging and has been widely used in relevant domains [7, 1, 4].
- presents the paper in the context of debugging AI specifications
- defines minimum unsatisfiable core
- presents dedicated approaches to compute MUC of  $LTL_f$  formulas for further diagnosis. There are several works on ex
- The main goal of this paper is to conduct an efficient MUC solver for unsatisfiable  $LTL_f$  formulas
- 

## Mine vs theirs

- Mention how  $LTL_f$  is different from MLTL and how that is an important distinction
- explain why MLTL cannot reduce to  $LTL_f$
- Also discuss how in-depth MLTL has been studied similar to how they did
- Talk about specification debugging and use the same references, but add more! :)

## Towards a notion of unsatisfiable and unrealizable cores for LTL [6]

- notions of unsatisfiable cores for LTL that arise from the syntax tree of an LTL formula
- unsatisfiable cores for LTL by converting to Conjunctive normal form
- unsatisfiable cores for LTL by proofs of its unsatisfiability

## **IEEE Recommended Practice for Software Requirements Specifications [1]**

Software requirements specification best practices. Definition of satisfiability.

## **Specification debugging**

### **A Genetic Algorithm for Goal-Conflict Identification [2]**

mentions unsatisfiability in specifications for their systems

Identifying these circumstances early in the development process is of most importance, since it enables one to improve specifications, take countermeasures to these situations, and more deeply understand the roots for potential system goal unsatisfiability

### **Goal-Conflict Detection Based on Temporal Satisfiability Checking [3]**

Our tableau-based approach to automatically detect goal conflicts receives a goal-oriented requirements specification composed of LTL formulas capturing the domain assumptions Dom, as well as goals  $G = G1, \dots, Gn$ . The process may determine that there are no conflicts, or that there exist either strong or weak conflicts.

### **Debugging overconstrained declarative models using unsatisfiable cores [7]**

This information can be a great help in discovering and localizing overconstraint, and in some cases pinpoints it immediately.

### **How to Identify Boundary Conditions with Contrasty Metric? [4]**

The boundary conditions (BCs) have shown great potential in requirements engineering because a BC captures the particular combination of circumstances, i.e., divergence, in which the goals of the requirement cannot be satisfied as a whole.

## **References**

- [1] Standards Association et al. Ieee recommended practice for software requirements specifications. ieee std 830 1998, 2016.

- [2] Renzo Degiovanni, Facundo Molina, Germán Regis, and Nazareno Aguirre. A genetic algorithm for goal-conflict identification. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pages 520–531, 2018.
- [3] Renzo Degiovanni, Nicolás Ricci, Dalal Alrajeh, Pablo Castro, and Nazareno Aguirre. Goal-conflict detection based on temporal satisfiability checking. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pages 507–518, 2016.
- [4] Weilin Luo, Hai Wan, Xiaotong Song, Binhao Yang, Hongzhen Zhong, and Yin Chen. How to identify boundary conditions with contrasty metric? In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 1473–1484. IEEE, 2021.
- [5] Tong Niu, Shengping Xiao, Xiaoyu Zhang, Jianwen Li, Yanhong Huang, and Jianqi Shi. Computing minimal unsatisfiable core for LTL over finite traces. *Journal of Logic and Computation*, 34(7):1274–1294, October 2024.
- [6] Viktor Schuppan. Towards a notion of unsatisfiable and unrealizable cores for LTL. *Science of Computer Programming*, 77(7-8):908–939, July 2012.
- [7] Ilya Shlyakhter, Robert Seater, Daniel Jackson, Manu Sridharan, and Mana Taghdiri. Debugging overconstrained declarative models using unsatisfiable cores. In *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.*, pages 94–105. IEEE, 2003.