A One-Pass Tree-Shaped Tableau for LTL+Past

Nicola Gigante
University of Udine, Italy

joint work with **Angelo Montanari**
University of Udine, Italy
and **Mark Reynolds**
University of Western Australia, Australia

LPAR-21, May 8-12 2017, Maun, Botswana

Introduction

Linear Temporal Logic

Linear Temporal Logic (LTL) is a propositional modal logic interpreted over infinite, discrete, linear orders.

$\mathsf{X}\,\alpha$	α will be true at the next state.
$\alpha\,\mathcal{U}\,\beta$	β will eventually be true, and α always holds until then.
$\mathsf{F}\,\beta \equiv \top\,\mathcal{U}\,\beta$	β will eventually be true.
$\mathsf{G}\,\beta \equiv \neg\,\mathsf{F}\,\neg\beta$	β will always be true.

Augmenting LTL with past operators

LTL can be augmented with past modalities:

$\mathsf{Y}\,\alpha$	α was true at the previous state.
$\alpha\,\mathcal{S}\,\beta$	β has been true in the past, and α always held since then.
$\mathsf{P}\,\beta \equiv \top\,\mathcal{S}\,\beta$	β has been true in the past.
$\mathsf{H}\,\beta \equiv \neg\,\mathsf{P}\,\neg\beta$	historically, β has always been true.

Why? Past operators do not add expressive power to LTL, but they do allow to express many formulae more succinctly.

Note: formulae are satisfied if they hold at the first state.

LTL Satisfiability

LTL satisfiability is the problem of checking whether there exists a model that satisfies a given LTL formula.

- PSPACE-complete problem.
- Algorithmic solutions:
  - (Büchi) Automata-based
  - Tableau methods
  - Temporal resolution
  - Reduction to model checking
  - …

The satisfiability problem for LTL+P is still PSPACE-complete.

Why LTL satisfiability?

LTL is usually used to write specification in model checking, but other applications exist for the satisfiability problem:

- sanity checking of specifications
- temporal reasoning in AI
- …

Tableau methods for LTL satisfiability

Tableaux were among the first methods proposed to solve the LTL satisfiability problem:

- Early tableau methods were graph-shaped and multiple-pass (Wolper 1984).
- Subsequently, Schwendimann [Sch98] introduced a *single*-pass tableau with a tree-like shape (still a DAG).

A one-pass tree-shaped tableau method for LTL satisfiability was recently proposed.

Reynolds 2016

M. Reynolds. "A New Rule for LTL Tableaux." In: *Proc. of the 7th International Symposium on Games, Automata, Logics and Formal Verification.* GandALF 2016

A One-Pass Tree-Shaped Tableau for LTL

A one-pass tree-shaped tableau method for LTL satisfiability was recently proposed, and implemented in a tool.

Bertello et al. 2016

M. Bertello, N. Gigante, A. Montanari, and M. Reynolds. "Leviathan: A New LTL Satisfiability Checking Tool Based on a One-Pass Tree-Shaped Tableau." In: *Proc. of the 25th International Joint Conference on Artificial Intelligence.* IJCAI 2016

`http://www.github.com/corralx/leviathan`

A One-Pass Tree-Shaped Tableau for LTL

A one-pass tree-shaped tableau method for LTL satisfiability was recently proposed, and implemented in a tool.

- Purely tree-shaped rule-based search procedure.
- A single pass is sufficient to determine the acceptance of rejection of a given branch.
- Very simple structure, combining the simplicity of declarative tableaux with the efficiency of one-pass systems.
- Easy to extend!
- Easy to parallelize (work in progress)!

A One-Pass Tree-Shaped Tableau for LTL+P

In this paper we extended the method to support LTL+P:

- The extension can be done in a very modular way:
    - it respects the same one-pass tree-shaped structure.
    - new rules are added to the system, with old rules left completely unchanged.
- First evidence of how this tableau can be easy to extend to different logics.

How it works

How it works

The tableau for ϕ is a tree where each node is labeled by a set of formulae, with the root labeled with $\{\phi\}$.

- The formula starts in Negated Normal Form.
- At each step some rules are applied to a leaf, depending on the contents of the label, possibly generating new children for the current node.
- Some rules can accept a branch, others can reject it.
- If the complete tree contains at least an accepted branch, the formula is satisfiable.

Expansion rules

Expansion rules are applied to a node until no other expansion rule can be applied anymore:

- Boolean connectives handled just like in classical propositional tableau.

$$\begin{array}{ccc} & \{\alpha \vee \beta\} & \\ / & & \backslash \\ \{\alpha\} & & \{\beta\} \end{array} \qquad \begin{array}{c} \{\alpha \wedge \beta\} \\ | \\ \{\alpha, \beta\} \end{array}$$

Expansion rules

Expansion rules are applied to a node until no other expansion rule can be applied anymore:

- Common expansion rules handle temporal operators:

$$\{\alpha\,\mathcal{U}\,\beta\} \;\to\; \{\beta\} \;\mid\; \{\alpha, \mathsf{X}(\alpha\,\mathcal{U}\,\beta)\}$$

$$\{\mathsf{F}\,\beta\} \;\to\; \{\beta\} \;\mid\; \{\mathsf{X}\,\mathsf{F}\,\beta\}$$

$$\{\mathsf{G}\,\alpha\} \;\to\; \{\alpha, \mathsf{X}\,\mathsf{G}\,\alpha\}$$

β is called an eventuality.

Advancing to the next temporal step

Once the current state has been fully expanded, we proceed to the next temporal state by the STEP rule:

$$\{\ldots, \mathsf{X}\,\alpha, \ldots\}$$
$$\downarrow$$
$$\{\alpha\}$$

Contradictions

If a label contains contradictions, we reject the branch.

$$\{\ldots, p, \ldots, \neg p, \ldots\}$$

✗

Acceptance and contradictions

If a STEP rule results into an empty label, we're done: the branch is accepted.

$$\{\ldots, p, \neg q, r, \ldots\}$$
$$\downarrow$$
$$\{\}$$
$$\checkmark$$

Finding periodic models - LOOP rule

Some formulae (*e.g.*, $\mathsf{G}\,\mathsf{F}\,p$) require to satisfy infinitely often the same request, thus the labels may never become empty.

This formulae will have infinite periodic models:

LOOP rule

If two nodes $u < v$ with labels $\Gamma_u = \Gamma_v$ are found and all the eventualities in Γ_u are fulfilled inbetween, the branch is accepted and the model loops through u and v.

Example

$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$|$$

$$\{\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

Example

$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$

$\{\mathsf{F}(p \wedge \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$

$\ldots$

$\{p, \mathsf{X}\,\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$\ldots$

$$\{p,\ \mathsf{X}\,\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\,\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

Example

$$\{\mathrm{G}\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p)\}$$

$$\{\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p), \mathrm{X}\,\mathrm{G}\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p)\}$$

$$\ldots \qquad \{p,\ \mathrm{X}\,\neg p,\ \mathrm{X}\,\mathrm{G}\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathrm{G}\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p)\}$$

$$\{\neg p,\ \mathrm{F}(p \wedge \mathrm{X}\,\neg p),\ \mathrm{X}\,\mathrm{G}\,\mathrm{F}(p \wedge \mathrm{X}\,\neg p)\}$$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p,\ \mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p, p,\ \mathsf{X}\neg p, \ldots\}$$

✗

Example

$\{\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

$\{\mathsf{F}(p \land \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

$\ldots$ $\quad$ $\{p, \mathsf{X}\,\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

$\downarrow$

$\{\neg p, \mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

$\{\neg p, \mathsf{F}(p \land \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

$\{\neg p, p, \mathsf{X}\,\neg p, \ldots\}$ $\quad$ $\{\neg p, \mathsf{X}\,\mathsf{F}(p \land \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$

✗

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\,\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\neg p,\ \mathsf{F}(p \wedge \mathsf{X}\,\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\neg p, p,\ \mathsf{X}\,\neg p, \ldots\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$✗ \qquad \downarrow$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\,\neg p),\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\,\neg p)\}$$

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p,\ \mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p, p,\ \mathsf{X}\neg p, \ldots\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

✗ $\qquad \downarrow$

$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$$\{\mathsf{F}(p \land \mathsf{X}\,\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$\ldots$

$$\{p,\ \mathsf{X}\,\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$$\{\neg p,\ \mathsf{F}(p \land \mathsf{X}\,\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$$\{\neg p, p,\ \mathsf{X}\,\neg p, \ldots\}$$

✗

$$\{\neg p,\ \mathsf{X}\,\mathsf{F}(p \land \mathsf{X}\,\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$$\downarrow$$

$$\{\mathsf{F}(p \land \mathsf{X}\,\neg p),\ \mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

$\ldots$

$$\{p,\ \mathsf{X}\,\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \land \mathsf{X}\,\neg p)\}$$

Example

$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\downarrow$$

$$\{\neg p,\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p,\ \mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\{\neg p, p,\ \mathsf{X}\neg p, \ldots\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

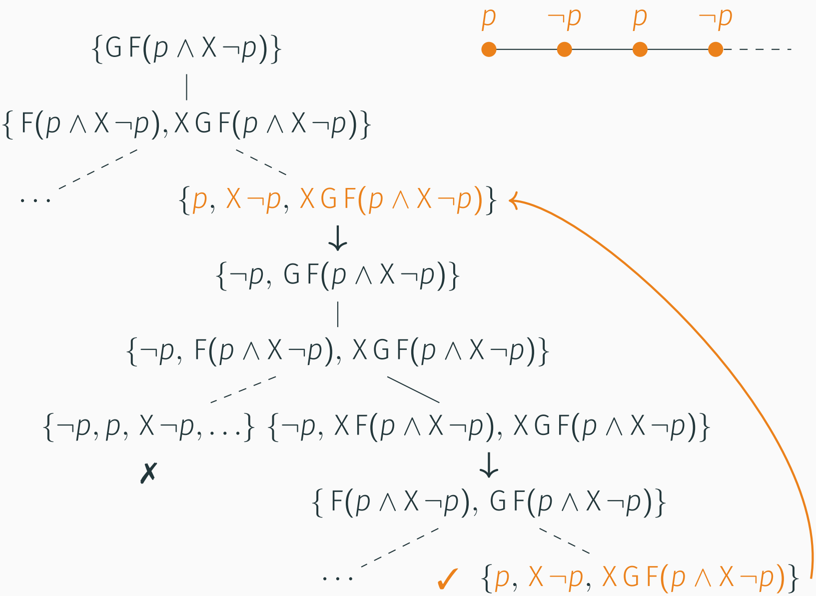
✗

$$\downarrow$$

$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p),\ \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

$$\ldots \qquad \{p,\ \mathsf{X}\neg p,\ \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$

Example


p $\neg p$ p $\neg p$
$\{\mathrm{G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\{\mathrm{F}(p \wedge \mathrm{X}\neg p), \mathrm{X\,G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\ldots$
$\{p, \mathrm{X}\neg p, \mathrm{X\,G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\{\neg p, \mathrm{G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\{\neg p, \mathrm{F}(p \wedge \mathrm{X}\neg p), \mathrm{X\,G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\{\neg p, p, \mathrm{X}\neg p, \ldots\}$
$\{\neg p, \mathrm{X\,F}(p \wedge \mathrm{X}\neg p), \mathrm{X\,G\,F}(p \wedge \mathrm{X}\neg p)\}$
✗
$\{\mathrm{F}(p \wedge \mathrm{X}\neg p), \mathrm{G\,F}(p \wedge \mathrm{X}\neg p)\}$
$\ldots$
✓ $\{p, \mathrm{X}\neg p, \mathrm{X\,G\,F}(p \wedge \mathrm{X}\neg p)\}$

Unrealizable eventualities

Something is still missing. Consider the following formula:

$$\mathsf{G}\,\neg p \land q\,\mathcal{U}\,p$$

- It is unsatisfiable, but not because of propositional contradictions.
- The requested eventuality is unrealizable.

Unrealizable eventualities - PRUNE rule

In these cases we have to stop postponing the eventuality to guarantee termination:

PRUNE rule

If three occurrences of the same label Γ are found in three nodes $u < v < w$ and the set of eventualities fulfilled between u and v is the same of those between v and w, the branch is rejected.

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$|$$

$$\{\,\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\,\mathsf{G}\,\neg p,\ q\,\mathcal{U}\,p\}$$

$$\{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ p\} \quad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\,\mathsf{G}\,\neg p,\ q\,\mathcal{U}\,p\}$$

$$\{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ p\} \quad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\,\mathsf{G}\,\neg p,\ q\,\mathcal{U}\,p\}$$

$$\{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ p\} \quad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p,\ q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

Example - unsatisfiable formula

$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$

$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$

$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\}$ ✗

$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$

$\downarrow$

$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$

Example - unsatisfiable formula

$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$

$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$

$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\}$ ✗ $\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$

$\downarrow$

$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$

$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\}$ ✗ $\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$

$\downarrow$

$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ $\downarrow$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \quad \{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, q, \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗ ✗

How it works - summary

To summarize:

- When to accept a branch?
  - When the label is empty
  - When we are looping while satisfying all the eventualities
- When to reject a branch?
  - When a label is contradictory
  - When we are looping but unable to satisfy all the eventualities

How it works - summary

To summarize:

- When to accept a branch?
  - When the label is empty
  - When we are looping while satisfying all the eventualities
- When to reject a branch?
  - When a label is contradictory
  - When we are looping but unable to satisfy all the eventualities

Supporting past operators

Supporting past operators

Handling the past is trivial in graph-shaped tableaux:

- Just build the graph edges such that each $\mathsf{Y}\alpha$ is satisfied

Our one-pass tableau is different:

- In each branch we are committed to a single history
- How to ensure the satisfaction of past requests if the past is fixed already?

Supporting past operators - expansion rules

Past temporal operators other than $\mathsf{Y}\,\alpha$ are expanded like their future counterparts:

$$\begin{array}{c} \{\alpha\,\mathcal{S}\,\beta\} \\ \swarrow \quad \searrow \\ \{\beta\} \qquad \{\alpha, \mathsf{Y}(\alpha\,\mathcal{S}\,\beta)\} \end{array} \qquad \begin{array}{c} \{\mathsf{P}\,\beta\} \\ \swarrow \quad \searrow \\ \{\beta\} \qquad \{\mathsf{Y}\,\mathsf{P}\,\beta\} \end{array} \qquad \begin{array}{c} \{\mathsf{H}\,\alpha\} \\ | \\ \{\alpha, \mathsf{Y}\,\mathsf{H}\,\alpha\} \end{array}$$

Thus the problem reduces to correctly handling $\mathsf{Y}\,\alpha$ formulae.

Supporting past operators - the YESTERDAY rule

Introducing the YESTERDAY rule:

- If u is such that $\mathsf{Y}\,\alpha \in \Gamma_u$ and the STEP rule has never been applied before, then the branch is rejected.
- Otherwise, let v be the node to which we lastly applied the STEP rule.
  - If we cannot find α in v nor in its expanded ancestors, then the branch is rejected.
  - A new child v' is added to v, with $\Gamma_{v'} = \Gamma_v \cup \{\alpha\}$

Supporting past operators - example

$$\{\phi\}$$

Supporting past operators - example

$$\{\phi\}$$

$$\{\ldots, \mathsf{X}\,\mathsf{Y}(p \lor q), \ldots\} \qquad \cdots$$

Supporting past operators - example

$$\{\phi\}$$

$$\{\ldots, \mathsf{X}\,\mathsf{Y}(p \lor q), \ldots\} \qquad \ldots$$

$$\downarrow$$

$$\{\ldots, \mathsf{Y}(p \lor q), \ldots\}$$

Supporting past operators - example

$$\{\phi\}$$

$$\{\ldots, \mathsf{X}\,\mathsf{Y}(p \vee q), \ldots\} \qquad \ldots$$

$$\downarrow$$

$$\{\ldots, \mathsf{Y}(p \vee q), \ldots\}$$

✗

Supporting past operators - example

$$\{\phi\}$$

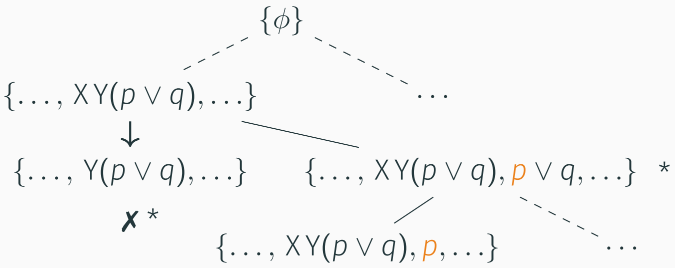
$\{\ldots, \mathsf{X}\,\mathsf{Y}(p \vee q), \ldots\}$ $\qquad \ldots$

$\downarrow$

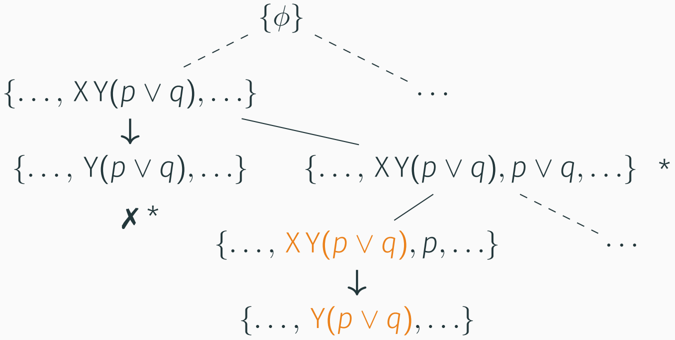
$\{\ldots, \mathsf{Y}(p \vee q), \ldots\}$ $\qquad$ $\{\ldots, \mathsf{X}\,\mathsf{Y}(p \vee q), p \vee q, \ldots\}$ *

✗ *

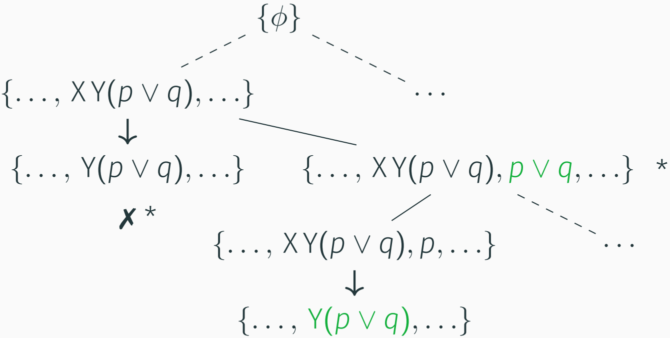
Supporting past operators - example



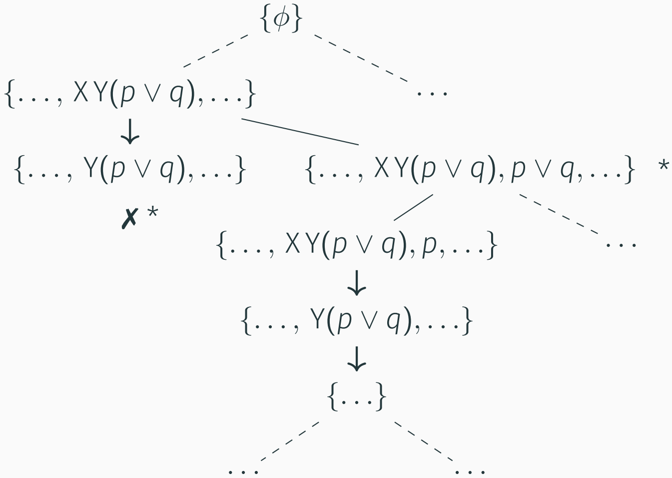
Supporting past operators - example



Supporting past operators - example



Supporting past operators - example



Conclusions

Conclusions

We extended a recent one-pass tree-shaped tableau method for LTL satisfiability to cover past operators:

We provided a very modular extension:

- The extension requires only a single new rule for each new temporal operator.
- We preserve the one-pass rule-based tree search structure of the procedure.
- We provide full soundness and completeness proofs:
  - soundness never appeared before (future-only neither)
  - improved, clarified completeness proof

Future work

Future lines of work:

- Add the past to our satisfiability checking tool.
  - Not trivial: our rule causes a lot of backtracking
- Exploit the modular structure of the tableau to extend it to other LTL extensions:
  - LTL on finite traces,
  - LTL with forgettable past,
  - metric extensions of LTL,
  - Alur & Hentzinger TPTL logic [AH94],
  - ...
- Implement these extensions: one tool for a broad family of linear time logics

Thank you!

Questions?

[AH94] Rajeev Alur and Thomas A. Henzinger. “A Really Temporal Logic.” In: *Journal of the ACM* (1994).

[Ber+16] M. Bertello, N. Gigante, A. Montanari, and M. Reynolds. “Leviathan: A New LTL Satisfiability Checking Tool Based on a One-Pass Tree-Shaped Tableau.” In: *Proc. of the 25th International Joint Conference on Artificial Intelligence*. IJCAI 2016.

[Rey16] M. Reynolds. “A New Rule for LTL Tableaux.” In: *Proc. of the 7th International Symposium on Games, Automata, Logics and Formal Verification*. GandALF 2016.

[Sch98] S. Schwendimann. "A New One-Pass Tableau Calculus for PLTL." In: *Proc. of the 7th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. TABLEAUX '98.