Two-Automaton Accepting Run Search in Spot

Clément Gillard

LRDE
EPITA Research and Development Laboratory

Seminar – 2018-07-03

Two-Automaton Accepting Run Search in Spot

Introduction

The model checking toolchain

The Model Checking Toolchain

$$M \models \varphi$$

The Model Checking Toolchain

$$
\begin{aligned}
& M \models \varphi \\
\iff \quad & L(A_M) \subseteq L(A_\varphi)
\end{aligned}
$$

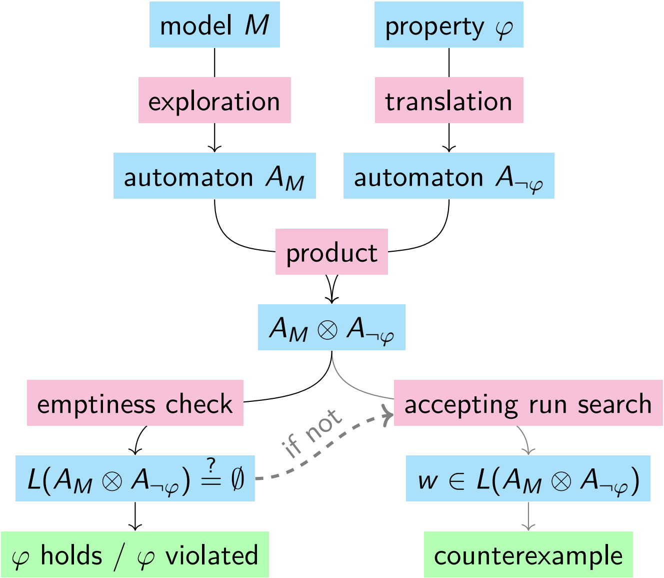
The Model Checking Toolchain

$$
\begin{aligned}
& M \models \varphi \\
\iff \quad & L(A_M) \subseteq L(A_\varphi) \\
\iff \quad & L(A_M \otimes \overline{A_\varphi}) = \emptyset
\end{aligned}
$$

The Model Checking Toolchain

$$
\begin{aligned}
& M \models \varphi \\
\iff\quad & L(A_M) \subseteq L(A_\varphi) \\
\iff\quad & L(A_M \otimes \overline{A_\varphi}) = \emptyset \\
\iff\quad & L(A_M \otimes A_{\neg\varphi}) = \emptyset
\end{aligned}
$$

The Model Checking Toolchain



The Spot library

Spot 2

- "Spot 2.0 — a framework for LTL and ω-automata manipulation" [4]
- C++ library
- algorithms, bridges for model checking
- executables for testing and comparing
  - `ltlcross`, `autcross`: language equivalence

The problem

The Problem

```
info: check_empty P1*N0
info: check_empty P1*N1
info: building Comp(N1)*Comp(P1) requires more acceptance sets than supported
info: building state-space #0/1 of 200 states with seed 0
info: state-space has 4225 edges
```

- ω-automata in Spot cannot have more than fixed number of acceptance sets (32 by default)
- acceptance sets sizes add-up during product

Operands may be expressed but not the product

Last Year's Work

Two-automaton emptiness check:

- Two-automaton emptiness check in Spot [5]
- Simulate a product on-the-fly
- Based on Couvreur's on-the-fly emptiness check algorithm [2]
- Optimisations on automata strength and API

No accepting run search, we cannot use it to look for a counterexample.

What we need to build

What We Need To Build

Pumping lemma: infinite word, but finite automata

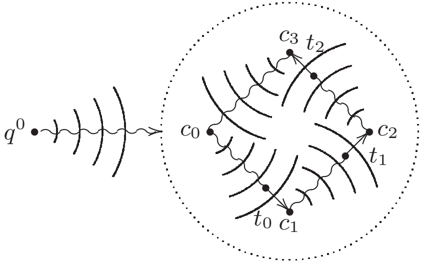
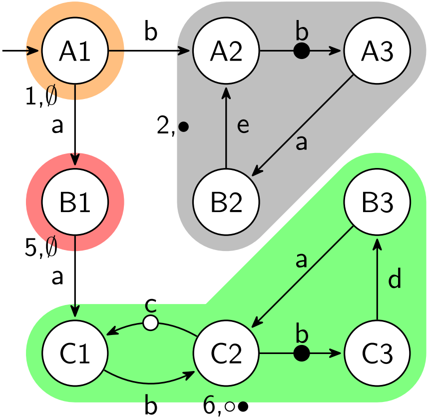


Fig. 5. Computing an accepting run for a TGBA.

Excerpt from “On-the-Fly Emptiness Checks for Generalized Büchi Automata” [3]

Example

Example



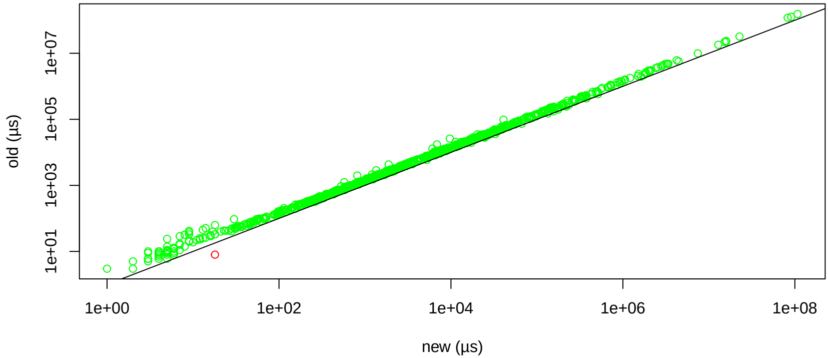
$$Inf(\bullet) \wedge Inf(\circ)$$

Results

Speed up

Speed Up

Comparison of old vs. new emptiness check in `ltlcross`



- 100 formulae[1] × 3 tools[2] ⇒ 849 emptiness checks (all empty, no accepting run search) + 27 timeouts
- outlier: "`(!(X(p11))) -> (!(p0))`" through `ltl3ba`

[1] `randltl --tree-size=30 30`

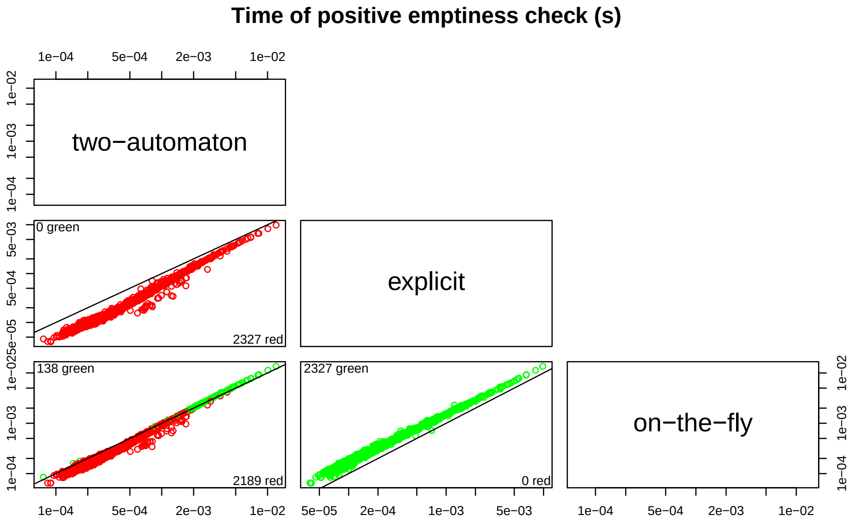
[2] `ltlcross -T 60 ltl2ba ltl3ba ltl2tgba`

Comparison with other implementations

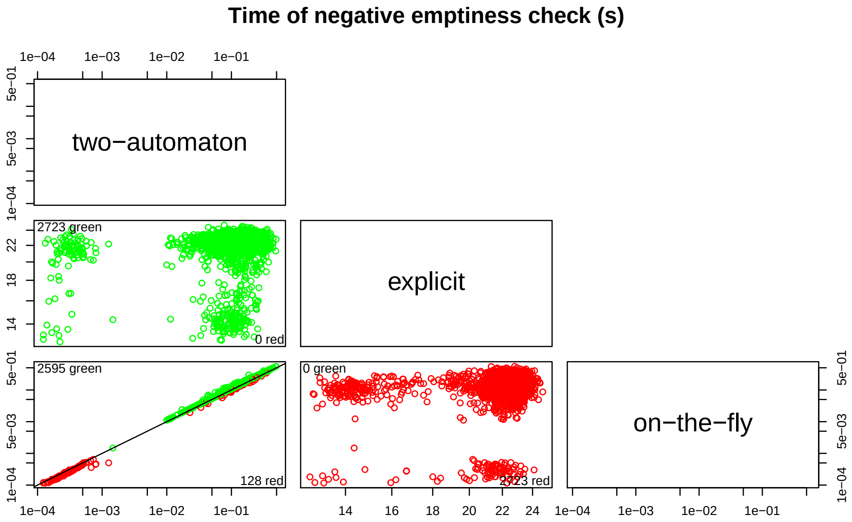
Bench Setup

1. We measure the time spent in:
   - two-automaton emptiness check
   - explicit product + emptiness check
   - on-the-fly product + emptiness check
   - + accepting run search when not empty
2. Data set:
   - 100 random automata: 200 states, 16 acceptance sets, 10 condition variables
   - 5050 combinations, 2668 empty, 2382 non-empty
3. In parallel over 24 cores of the same machine

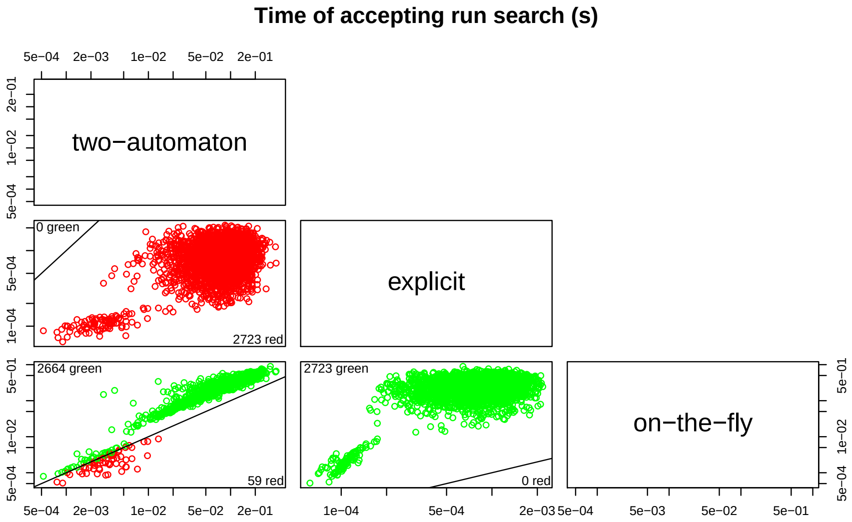
Case of Emptiness



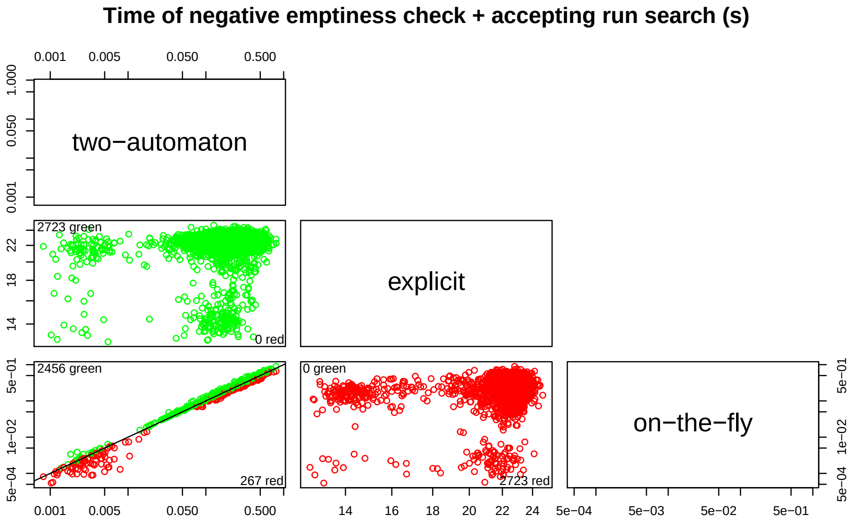
Case of Non-Emptiness



Case of Non-Emptiness



Case of Non-Emptiness



Conclusion

Conclusion

What has been done:

- Make the two-automaton emptiness check fully usable
- Test it on a broader range

What remains to be done:

- Investigate on performances
- From generalised Büchi to Rabin: “Explicit State Model Checking with Generalized Büchi and Rabin Automata” [1]

Questions?

Bibliography I

- Vincent Bloemen, Alexandre Duret-Lutz, and Jaco van de Pol. "Explicit State Model Checking with Generalized Büchi and Rabin Automata". In: Proceedings of the 24th International SPIN Symposium on Model Checking of Software (SPIN'17). ACM, July 2017, pp. 50–59.
- Jean-Michel Couvreur. "On-the-fly Verification of Temporal Logic". In: Proceedings of the World Congress on Formal Methods in the Development of Computing Systems (FM'99). Ed. by Jeannette M. Wing, Jim Woodcock, and Jim Davies. Vol. 1708. Lecture Notes in Computer Science. Toulouse, France: Springer-Verlag, Sept. 1999, pp. 253–271. ISBN: 3-540-66587-0.

Bibliography II

Jean-Michel Couvreur, Alexandre Duret-Lutz, and Denis Poitrenaud. "On-the-Fly Emptiness Checks for Generalized Büchi Automata". In: Proceedings of the 12th International SPIN Workshop on Model Checking of Software (SPIN'05). Ed. by Patrice Godefroid. Vol. 3639. Lecture Notes in Computer Science. Springer, Aug. 2005, pp. 143–158.

Alexandre Duret-Lutz et al. "Spot 2.0 — a framework for LTL and ω-automata manipulation". In: Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis (ATVA'16). Vol. 9938. Lecture Notes in Computer Science. Springer, Oct. 2016, pp. 122–129.

Bibliography III

Clément Gillard. Two-automaton emptiness check in Spot. Tech. rep. 1706. EPITA Research and Development Laboratory (LRDE), 2017.