Spot

Feedback on the development of this 15-year-old project

Alexandre Duret-Lutz
LRDE/EPITA

4 April 2019

Current State

A platform for LTL, ω-automata manipulation, and model checking.

Current State

A platform for LTL, ω-automata manipulation, and model checking.

Distribution (under GPLv3+):

- C++14 library (~100 KLOC)
- 14 command-line tools (~13 KLOC) ▸ tools
- bindings for Python 3, support for Jupyter notebooks
- large test-suite (~55 KLOC, ~90% coverage)
- different types of documentation (not enough) ▸ doxygen ▸ tut ▸ TL
- Debian packages, RPM packages

Current State

A platform for LTL, ω-automata manipulation, and model checking.

Distribution (under GPLv3+):

- C++14 library (~100 KLOC)
- 14 command-line tools (~13 KLOC) ▸ tools
- bindings for Python 3, support for Jupyter notebooks
- large test-suite (~55 KLOC, ~90% coverage)
- different types of documentation (not enough) ▸ doxygen ▸ tut ▸ TL
- Debian packages, RPM packages

Additional support:

- web app for LTL translation/comparison/exploration ▸ app
- online sandbox for testing tools and Python bindings ▸ sandbox
- two mailing lists

Notable features (in 2019)

- Support ω-automata with transition-based acceptance ▸ example
- Acceptance conditions are formulas (EL-style) ▸ example
- Many ω-automata algorithms that work for any acceptance condition. (Usually improved w.r.t. original publication.)
- Automaton parser can read 4 syntaxes, and supports streaming.
- Support for a linear fragment of PSL (= LTL + semi-extended regexes).
- Good translation from LTL to BA/GBA/EL. ▸ bench 2017
- Command-line tools designed with piping in mind, coherent set of options, and following GNU Coding Standards. ▸ GCS
- About 10 releases every year.

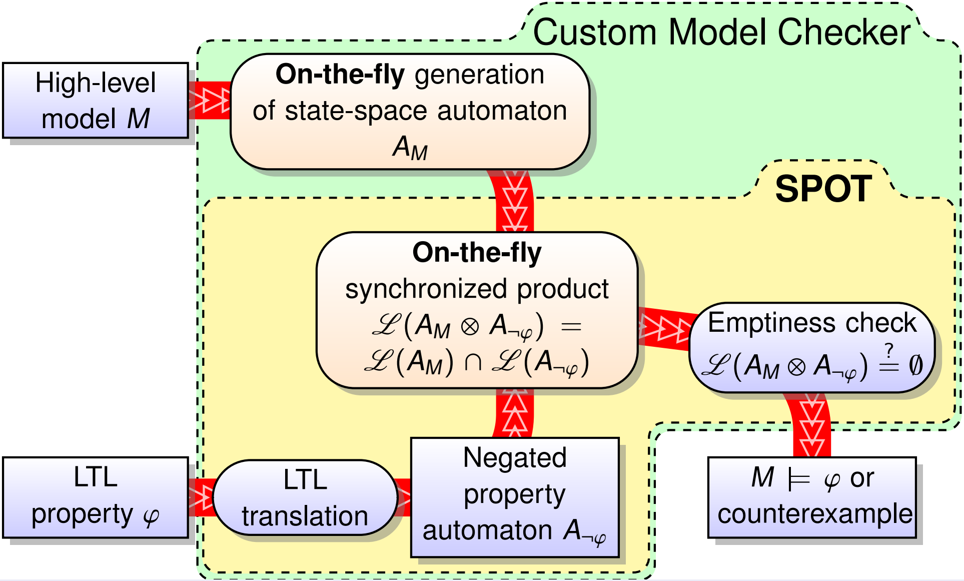
Evolution — Spot 0.x

2003–2007 "C++ model-checking library based on TGBAs"

- library only, no binaries for users
- minimal Python bindings, mainly used for web LTL translation
- focus on using an on-the-fly interface

A. Duret-Lutz and D. Poitrenaud. SPOT: an Extensible Model Checking Library using Transition-based Generalized Büchi Automata. *MASCOTS'04*

Automata-Theoretic LTL Model Checking


Custom Model Checker
High-level model M
On-the-fly generation of state-space automaton A_M
SPOT
On-the-fly synchronized product $\mathscr{L}(A_M \otimes A_{\neg\varphi}) = \mathscr{L}(A_M) \cap \mathscr{L}(A_{\neg\varphi})$
Emptiness check $\mathscr{L}(A_M \otimes A_{\neg\varphi}) \overset{?}{=} \emptyset$
LTL property φ
LTL translation
Negated property automaton $A_{\neg\varphi}$
$M \models \varphi$ or counterexample

A. Duret-Lutz and D. Poitrenaud. SPOT: an Extensible Model Checking Library using Transition-based Generalized Büchi Automata. *MASCOTS'04*

Evolution — Spot 0.x

2003–2007 "C++ model-checking library based on TGBAs"

- library only, no binaries for users
- minimal Python bindings, mainly used for web LTL translation
- focus on using an on-the-fly interface

> *Among all the tools we tested, only SPOT can be considered an industrial quality tool. [...] SPOT, the best LTL translator in our experiments [...]*
>
> C. Rozier and M. Vardi, 2007

Benchmarks use a binary called `ltl2tgba` found in the test-suite, with a **crappy** interface. (Default options give the worst output.)

Evolution — Spot 0.x phase 2

2007–2011 "C++ model-checking library based on TGBAs" (still)

- library only, no binaries for users
- minimal Python bindings, still for web LTL translation (v2)
- some optimizations based on use-cases seen in papers
- PSL support, WDBA-minimization, automata simplifications
- some work at building a model checker on top of Spot ▸ its-ltl

Evolution — Spot 0.x phase 2

2007–2011 "C++ model-checking library based on TGBAs" (still)

- library only, no binaries for users
- minimal Python bindings, still for web LTL translation (v2)
- some optimizations based on use-cases seen in papers
- PSL support, WDBA-minimization, automata simplifications
- some work at building a model checker on top of Spot ▸ its-ltl

Our test-binary `ltl2tgba` now supports an impressive number of options, each triggering a specific algorithm. Getting small automata out of `ltl2tgba` (or in C++) requires some know-how.

```
tests/ltl2tgba -f -r7 -R3 -Rm -RDS -x 'φ'
```

▸ bench 2012

Some papers introduce new LTL translation algorithms and compare them against Spot using `ltl2tgba` with default options. This corresponds to the worst algorithm available in Spot...

Evolution — Spot 1.x

2012–2013 "C++ model-checking library + command-line tools"

- command-line tools introduced with sane user interface
  - Follow GCS for handling arguments. ▸ GCS
  - Uniform interface between tools.
  - Built with piping/batch processing in mind.
  - Restricted to LTL input.
    (Because no streamable format for automata.)
  - Support for working with CSV files. ▸ CSV
- `bin/ltl2tgba` user-friendly replacement for `tests/ltl2tgba`:

  ```
  bin/ltl2tgba --small 'φ'
  ```

  Yet some continue to use `tests/ltl2tgba` in publications...
- Improvements to LTL translation. A C++ class `translator` hides the logic of selecting appropriate algorithms.
- `bin/dstar2tgba` introduced to work on SAT-based minimization for deterministic TGBA. Parser for DSTAR. ▸ dstar2tgba

Evolution — Spot 2.x

2013– "C++ library for ω-automata and LTL manipulation"

- **Major rewrite** to support the HOA format. ▸ HOA
- Command-line tools augmented with tools that process automata.
- Shift in focus: model checking is one application, not a goal. LTL synthesis is another application.
- Python bindings are more complete. Jupyter notebooks good for reproductive research, experimenting, teaching, and also developing.
- Multiple third-party tools are now built above Spot. (Versioning is a new issue.)

Complaints

- Modularity (ATVA'16 reviewer)

> *The only complain is, Spot 2.0 now seems to have too many functionalities to handle now. [...] consider to decompose Spot 2.0? [...] Spot is famous for LTL-to-BA translation, why not separate this part with others such that users can focus on what they really want?*

- License: GPL is viral (Tina man page)

> ```
> -ltl2ba [...]  ltl2ba [...]  bundled with tina
> -spot [...]  ltl2tgba must be installed
> ```

- Documentation (some user on the mailing list)

> [...] very difficult to find in the documentation [...] how to use accepting states in a Büchi automaton.