

TChecker & the WP3

Frédéric Herbreteau
(LaBRI, Bordeaux INP)

April 4th, 2019
ANR TickTac Kickoff meeting, Rennes

WP3 in a nutshell

From our proposal, § 2.2:

*(WP3) Tool development: Workpackage WP3 is the core of our project, it consists in the **development of a state-of-the-art open-source tool for the verification of timed automata***

Integrate existing tools:

- ▶ Symrob (O. Sankur)
- ▶ TiAMo (M. Colange)
- ▶ and TChecker (F. Herbreteau, G. Point)

WP3 Goals

- ▶ **promote methods and algorithms** developed in the context of the project;
- ▶ **foster synergies between consortium members** through a common implementation platform;
- ▶ **encourage reproducible research** by providing in a single place all algorithms, benchmarks and experiments used throughout the project;
- ▶ **provide building blocks** (such as parsers and data structures, but also state-of-the-art algorithms) to facilitate the implementation of new algorithms;
- ▶ **offer a common benchmarking platform for experiments** and algorithms comparison

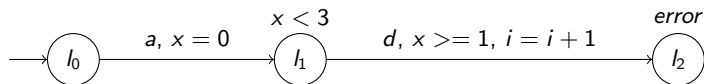
TChecker

What is TChecker?

- ▶ **Experimentation platform** for timed verification algorithms
- ▶ written in C++17 (47k lines including comments and spaces)
- ▶ distributed under MIT license (to be discussed)
- ▶ Repository: <https://github.com/fredher/tchecker.git>
- ▶ First public release April 1st, 2019 (3rd version)

Specifying timed automata in TChecker

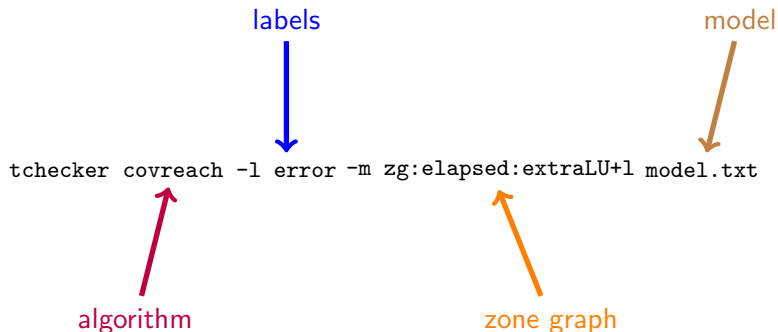
```
event:a
event:d
clock:1:x
int:1:0:9:0:i
process:P1
location:P1:l0{initial:}
location:P1:l1{invariant: x<3}
location:P1:l2{labels: error}
edge:P1:l0:l1:a{do: x=0}
edge:P1:l1:l2:d{provided: x>=1 : do: i=i+1}
```



Easily extensible using **attributes**.

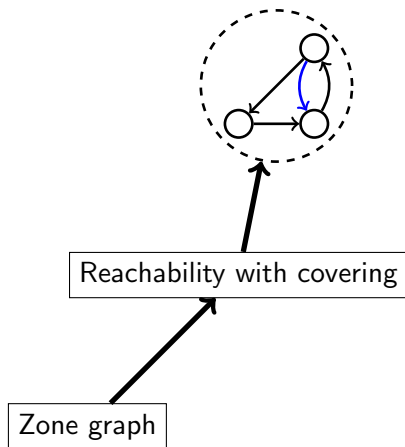
TChecker in action

Run reachability algorithm with covering:

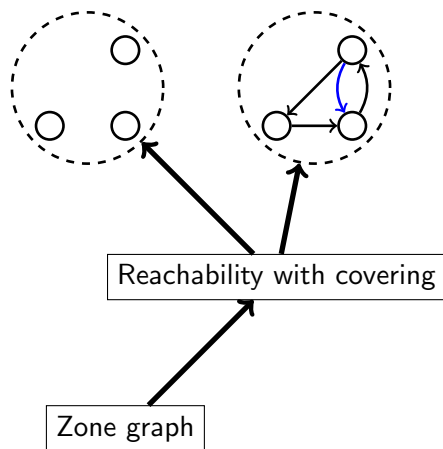


Demo: `reach-error.sh`, `reach-foo.sh`

Example of modularity in TChecker

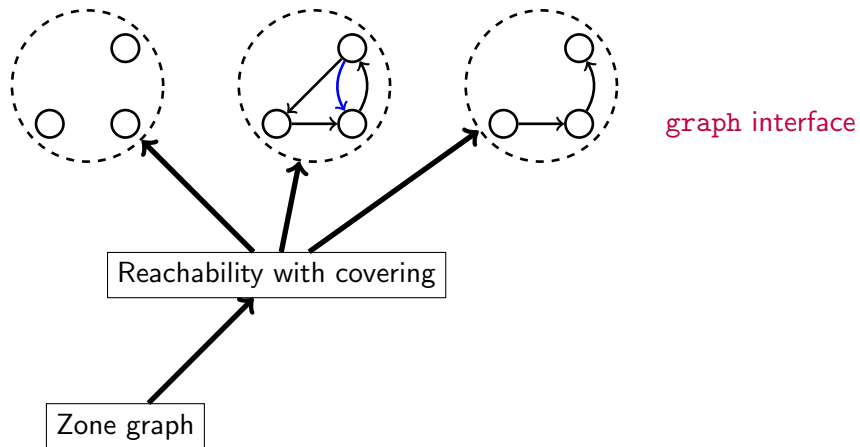


Example of modularity in TChecker

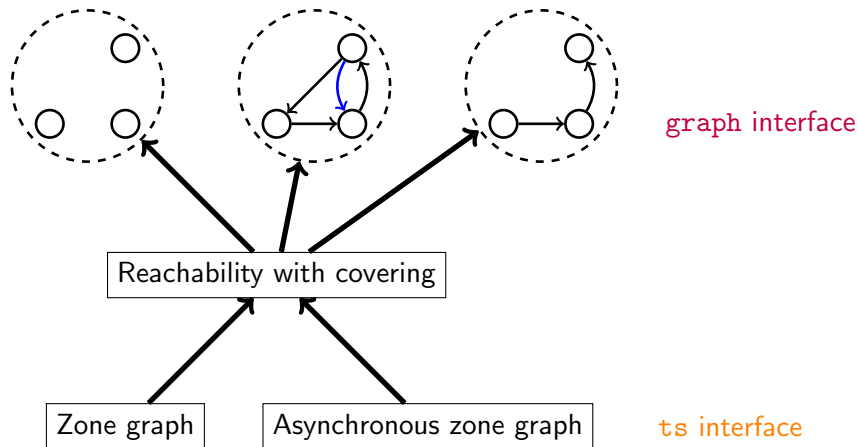


graph interface

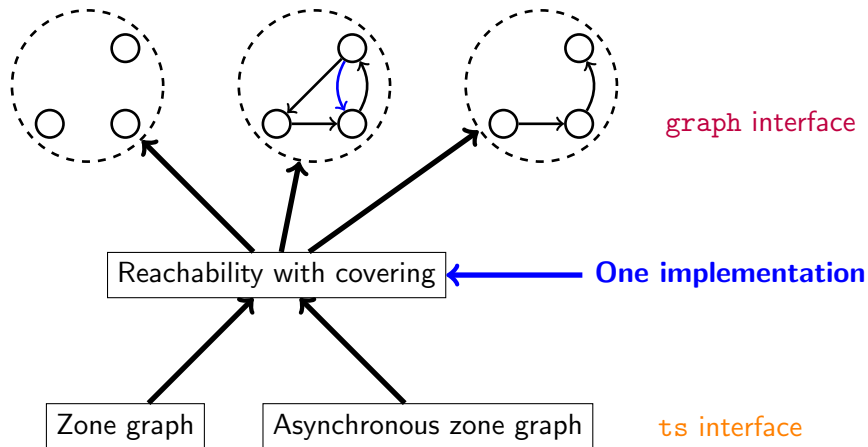
Example of modularity in TChecker



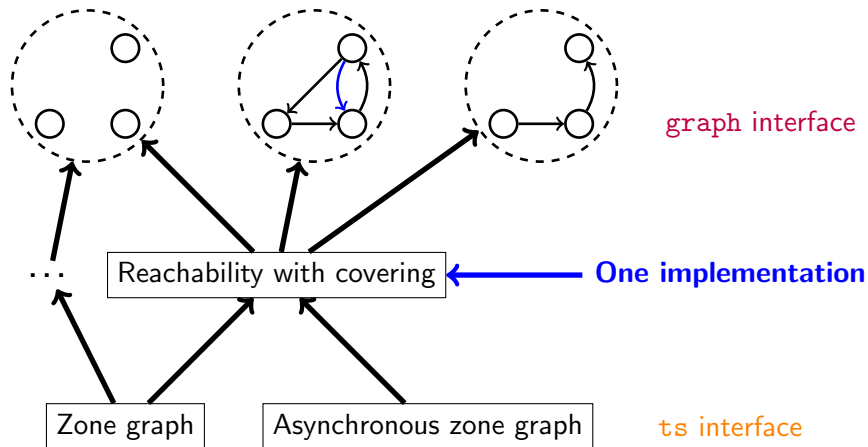
Example of modularity in TChecker



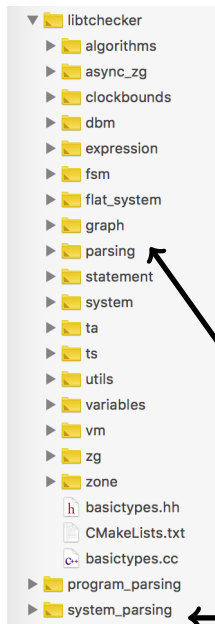
Example of modularity in TChecker



Example of modularity in TChecker

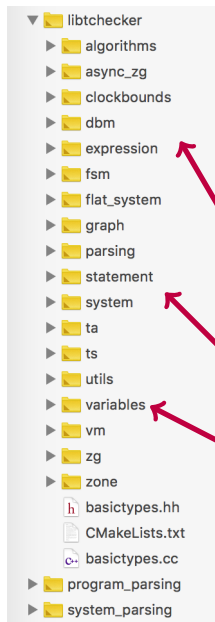


TChecker is a tool + a library



Parsing

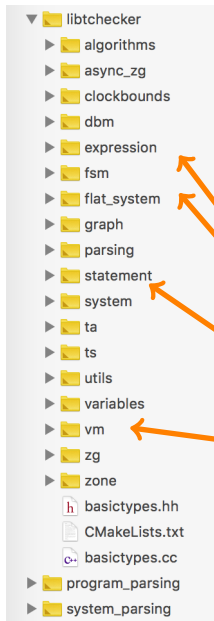
TChecker is a tool + a library



System representation

Parsing

TChecker is a tool + a library

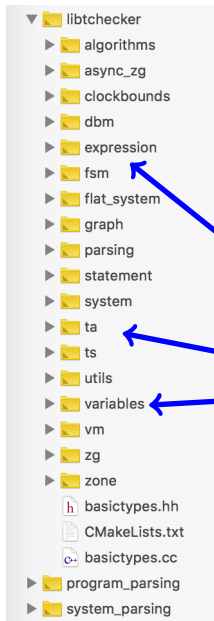


Finite state machine

System representation

Parsing

TChecker is a tool + a library



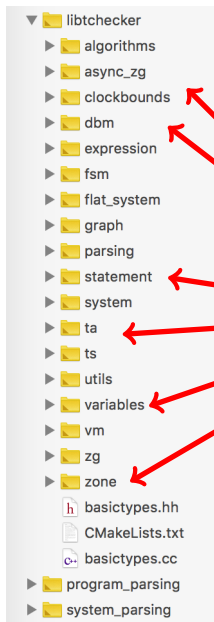
Timed automaton

Finite state machine

System representation

Parsing

TChecker is a tool + a library



Zone graphs

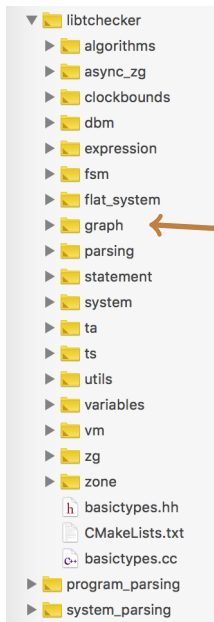
Timed automaton

Finite state machine

System representation

Parsing

TChecker is a tool + a library



Graphs

Zone graphs

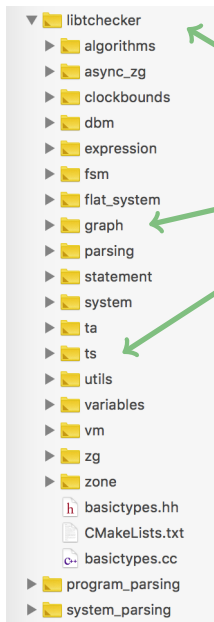
Timed automaton

Finite state machine

System representation

Parsing

TChecker is a tool + a library



Algorithms

Graphs

Zone graphs

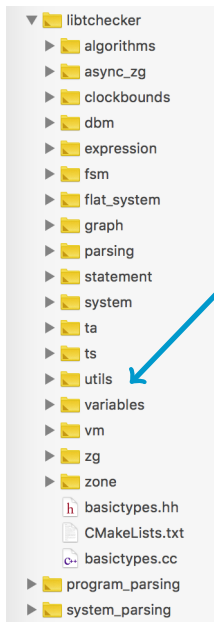
Timed automaton

Finite state machine

System representation

Parsing

TChecker is a tool + a library



Memory management

Algorithms

Graphs

Zone graphs

Timed automaton

Finite state machine

System representation

Parsing

Achieving WP3 goals

Tasks of WP3

- ▶ A first prototype to be delivered early in the project (**Year 1**), **aggregating the features of the tools developed by the members of the project.**
- ▶ **A website for our tool (Year 1)** which contains the source code, executables, bug reports, and the benchmark database gathering use-cases from the literature and from past experiments.
- ▶ **Extending the prototype to increase modularity.** This continuous task includes the support of various representation strategies (e.g. BDD, state caching, state compression), of various algorithmic solutions (e.g. support of abstraction/refinement procedures), or support of parallelism.
- ▶ The development of a **graphical interface for our tool (Year 4).**

Deliverables

- ▶ D3.1: Tool website, first prototype, and benchmark database (M12)
- ▶ D3.2: First stable release (M36)

Participants: Ocan, Frédéric, Gérald, Alexis, . . .

Next steps (M12)

- ▶ WP3 meeting (soon)
- ▶ Implement missing algorithms in TChecker, optimizations
- ▶ Integrate Symrob, TiAMo and TChecker
- ▶ UPPAAL to TChecker compiler