ATAC: A Tool for Automating Timed Automata Construction

Beyazit Yalcinkaya
Advisor: Ebru Aydin Gol
Department of Computer Engineering, Middle East Technical University

ODTÜ METU

Introduction

- Cyber-Physical Systems (CPS) are getting increasingly more attention from the research community as well as the industry.
- Systems with timing requirements, i.e., real-time systems (RTS), are an essential part of CPS in general.
- Timed Automata (TA) is a modelling formalism for RTS.
- Designing TA models is tedious and error-prone.

Timed Automata

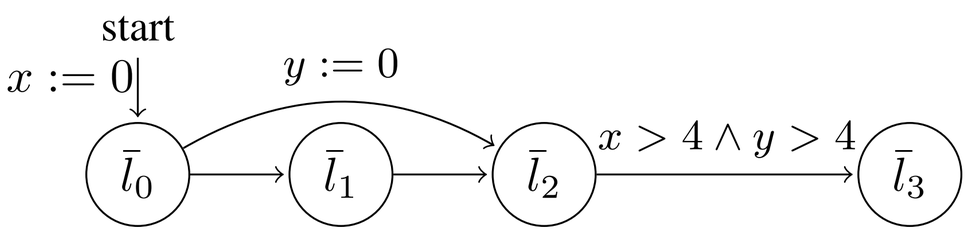

Figure 1: A TA model.

- TA is basically a finite state machine extended with a set of clock variables which monotonically increase.
- Clocks are real-valued counters used for modelling timing behaviour of a system.

UPPAAL

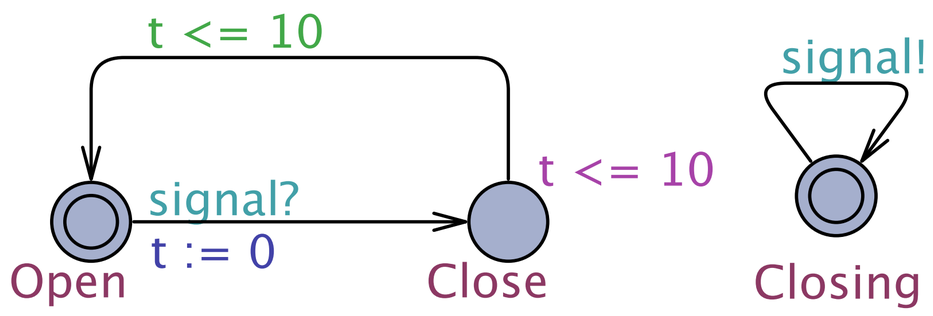

Figure 2: An UPPAAL model.

- A tool for modelling, designing, and verifying TA models.
- UPPAAL uses XML files to represent TA models.

Problem

- Develop a tool that automates TA design.
- Given descriptions and specifications in structured natural language, the tool should generate TA models that can be viewed and verified by UPPAAL.

Method

1. Define a formal grammar to identify a set of sentences.
2. Implement the tool as a Python program.
3. Make the tool online available.

Grammar

$$
\begin{aligned}
\phi_{TA} &::= \phi_{init}\ \phi_{main} \\
\phi_{main} &::= \phi_{sys}\ \phi_{main} \mid \phi_{spec}\ \phi_{main} \mid \epsilon \\
\phi_{init} &::= \textbf{temp}\ \textit{can only be}\ \textbf{loc} \\
&\quad \mid \textbf{temp}\ \textit{can be}\ \phi_{locs}\ \textit{and it is initially}\ \textbf{loc} \\
\phi_{sys} &::= \phi_{tran} \mid \textit{if}\ \phi_{cond}\ \textit{then}\ \phi_{tran} \\
\phi_{spec} &::= \textit{it goes to}\ \textbf{loc}\ \textit{in every}\ \textbf{N} \\
&\quad \mid \textit{the time spent in}\ \textbf{loc}\ \textit{cannot be more than}\ \phi_{eq}\ \textbf{N} \\
&\quad \mid \textbf{loc}\ \textit{must be reached from}\ \textbf{loc} \\
&\quad \mid \textbf{loc}\ \textit{must be reached from}\ \textbf{loc}\ \textit{within}\ \textbf{N}
\end{aligned}
$$

Figure 3: Main Grammar Rules.

ATAC

Light can be Off Dim Bright and it is initially Off.
If it receives press, then it can go to Dim from Off.
If it receives press and the time spent after leaving Off is less than 2, then it can go to Bright from Dim.
If it receives press and the time spent after leaving Off is more than or equal to 2, then it can go to Off from Dim.
If it receives press, then it can go to Off from Bright.
Presser can only be Pressing.
It can send press and go to Pressing from Pressing.

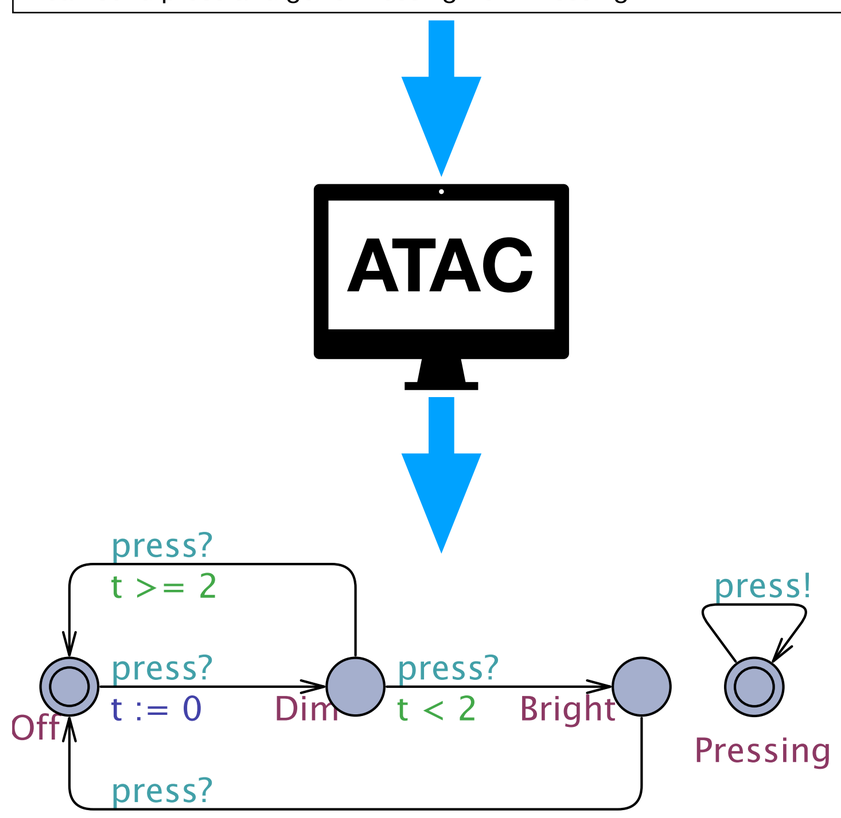

Figure 4: ATAC.

Conclusion

- ATAC extracts implied timing, synchronization, and transition description as well as any implied specification the system has to follow, then constructs a TA model following these descriptions and creates queries for checking against specifications.
- The generated models and formal specifications can be imported to UPPAAL for verification.

References

[1] Rajeev Alur. *Principles of cyber-physical systems*. MIT Press, 2015.

[2] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[3] Ben Caldwell. Pyuppaal. `https://github.com/bencaldwell/pyuppaal`, 2015.

[4] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[5] METU Cyber-Physical Systems Research Group. Atac: A tool for automating timed automata construction. `https://cps.ceng.metu.edu.tr/atac/`, 2019.

[6] Uppsala University and Aalborg University. Uppaal, 2005.

[7] Lark Parsing Library & Toolkit. Lark parser. `https://github.com/lark-parser/lark`, 2017.

[8] Beyazit Yalcinkaya and Ebru Aydin Gol. Clock reduction in timed automata while preserving design parameters. In *Proceedings of the 7th Conference on Formal Methods in Software Engineering*, page to appear. IEEE/ACM, 2019.