Bibliography

[1] Alur, R., Courcoubetis, C. and Dill, D. 1990. Model-checking for real-time systems. *Proceedings of the fifth annual IEEE symposium on logic in computer science*. IEEE. 414–425.

[2] Alur, R. and Dill, D.L. 1994. A theory of timed automata. *Theoretical computer science*. Elsevier. 126, 2. 183–235.

[3] Behrmann, G., David, A. and Larsen, K.G. 2004. A tutorial on Uppaal. *Formal methods for the design of real-time systems: Fourth international school on formal methods for the design of computer, communication, and software systems*. Springer. 200–236.

[4] Behrmann, G., David, A. and Larsen, K.G. A tutorial on Uppaal 4.0: *https://www.it.uu.se/research/group/darts/papers/texts/new-tutorial.pdf*. Accessed: 2020-09-21. Revised and extended version of [3].

[5] Bengtsson, J. and Yi, W. 2003. Timed automata: Semantics, algorithms and tools. *Advanced course on petri nets*. Springer. 87–124.

[6] Bouyer, P. et al. 2018. Model checking real-time systems. *Handbook of model checking*. Springer. 1001–1046.

[7] Boyd, S.P. Lecture notes on feedback control systems: *https://stanford.edu/~boyd/ee102/ctrl-static.pdf*. Accessed: 2021-01-05.

[8] Broman, D. 2019. A vision of Miking: Interactive programmatic modeling, sound language composition, and self-learning compilation. *Proceedings of the 12th ACM SIGPLAN international conference on software language engineering*. ACM. 55–60.

[9] Broman, D. High-confidence formal verification of real cyber-physical systems: From models to machine code: *https://people.kth.se/~dbro/projects.html*. Accessed: 2020-09-14.

[10] Cascading style sheets: *https://www.w3.org/Style/CSS*. Accessed: 2021-01-02.

[11] Daws, C. et al. 1995. The tool Kronos. *Lecture notes in computer science: Hybrid systems III, verification and control*. Springer. 208–219.

[12] Extensible markup language (XML) 1.0 (fifth edition): Notation: *https://www.w3.org/TR/REC-xml/#sec-notation*. Accessed: 2021-01-22.

[13] Graphviz: *https://graphviz.org*. Accessed: 2021-01-02.

[14] Graphviz: The Dot language: *https://graphviz.org/doc/info/lang.html*. Accessed: 2020-09-30.

[15] Henzinger, T.A. et al. 1994. *Symbolic model checking for real-time systems.* Cornell University.

[16] Jard, C. and Roux, O.H. 2009. *Communicating embedded systems.* Wiley.

[17] Kronos: *http://www-verimag.imag.fr/DIST-TOOLS/TEMPO/kronos*. Accessed: 2020-10-28.

[18] Lamport, L. 1987. A fast mutual exclusion algorithm. *ACM Transactions on Computer Systems.* ACM. 5, 1. 1–11.

[19] Menghi, C. et al. 2020. Model checking MITL formulae on timed automata: A logic-based approach. *ACM Transactions on Computational Logic.* ACM. 21, 3. 1–44.

[20] Miking: README: *https://github.com/miking-lang/miking/blob/develop/README.md*. Accessed: 2020-12-08.

[21] Moeller, M.O. 2001. Uppaal CSMA/CD model: *https://www.it.uu.se/research/group/darts/uppaal/benchmarks/csma/csma_input_02.ta*. Accessed: 2020-12-11.

[22] Nutt, G. and Bayer, D. 1982. Performance of CSMA/CD networks under combined voice and data loads. *IEEE Transactions on Communications.* IEEE. 30, 1. 6–11.

[23] Pombrio, J. and Krishnamurthi, S. 2014. Resugaring: Lifting evaluation sequences through syntactic sugar. *Proceedings of the 35th ACM SIGPLAN conference on programming language design and implementation.* ACM. 361–371.

[24] Uppaal: *http://www.uppaal.org*. Accessed: 2020-09-30.

[25] Uppaal 4.1.24 for 64-bit macOS: *http://uppaal.org/download/*. Accessed: 2020-12-02.

[26] Uppaal Tiga: *https://people.cs.aau.dk/~adavid/tiga/*. Accessed: 2020-10-27.

[27] Waez, M.T.B., Dingel, J. and Rudie, K. 2013. A survey of timed automata for the development of real-time systems. *Computer Science Review.* Elsevier. 9. 1–26.

[28] Yovine, S. 1997. Kronos: A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*. Springer. 1, 1-2. 123–133.

Appendix A

Intermediate Representation

This appendix details the intermediate JSON representation of each TML variant. The IR for TML-TSA is presented first, after which follows modifications used for the other language variants. For details on the JSON format, see `https://www.json.org`.

TML-TSA

Table A.1: Root object

Property	Type	Description
`actions`	Array of strings	All actions used in the program.
`clocks`	Array of strings	All clocks used in the program.
`locations`	Array of *Location* objects	Automaton locations.
`edges`	Array of *Edge* objects	Automaton edges.

Table A.2: Location object

Property	Type	Description
`id`	String	Location identifier.
`initial`	Boolean	True for the initial location, false for all others.
`invariant`	Null or string	Location invariant (optional).