

Analysis of of Partial Modeling Formalisms in Abstract Model Checking

Ou Wei¹, Arie Gurfinkel², and Marsha Chechik³

¹Nanjing University of Aeronautics and Astronautics

²Software Engineering Institute, Carnegie Mellon University

³Department of Computer Science, University of Toronto

Abstract Abstraction is the key to scaling model checking to industrial-sized problems. Typically, a large (or infinite) concrete system is approximated by a smaller abstract system via: (a) abstracting the concrete states, (b) analyzing the resulting abstract system, and (c) lifting the result back to the concrete system. Two common abstraction schemes are *over-approximation* – the abstract system contains *more* behaviours than the concrete one, and *under-approximation* – the abstract system contains *less* behaviours than the concrete one. Over-approximation is sound for universal properties (e.g., absence of errors). Under-approximation is sound for existential properties (e.g., presence of errors).

Abstractions that are sound for arbitrary properties, such as full μ -calculus L_μ , must combine over- and under-approximation into a *single* model [11, 3]. This leads to transition systems (TSs) with two types of transitions, *may* and *must*, representing *possible* (or over-approximating), and *necessary* (or under-approximating) behaviours, respectively. We call such systems *partial*. A temporal property is interpreted over a partial TS in one of four ways: *true* or *false*, if the partial TS is precise enough to prove or disprove the property, *unknown*, if the TS is imprecise, and *inconsistent* otherwise.

There are three families of partial modeling formalisms identified in the literature:

1. *Kripke Modal Transition Systems* (KMTSs) [9] and their equivalent variants, *Modal Transition Systems* (MTSs) [11], *Partial Kripke Structures* (PKSs) [1], and *3-valued Kripke Structures* [2]. KMTSs require that every *must* transition is also a *may* transition. They were introduced as computational models for partial specifications of reactive systems [11] and then adapted for model checking [1, 9, 2].
2. *Mixed Transition Systems* (MixTSs) [3], and equivalently, *Belnap Transition Systems* [7]. MixTSs extend KMTSs by allowing *must only* transitions (i.e., transitions that are *must* but not *may*). MixTSs were introduced in [3] as abstract models for L_μ , and have been used for predicate abstraction and software model checking in [6].
3. *Generalized KMTSs* (GKMTSs) [13], and equivalently, *Abstract TSs* [5] and *Disjunctive MTSs* [12]. GKMTSs extend MixTSs by allowing *must hyper-transitions*, (i.e., transitions into sets of states).

In this report, we present our work on analysis of these formalisms from two points of view [14, 15]: a semantic one, using partial TSs as objects for abstracting concrete systems, and a logical one, using partial TSs for temporal logic model checking. A partial TS is *semantically consistent* if it abstracts at least one concrete system. A partial

TS is *logically consistent* if it gives consistent interpretation to all temporal logic formulas. For semantic consistency, we investigate partial transition systems for abstract model checking, where a partial transition system and its concrete refinement are related through the soundness relation of abstract and concrete states. Specifically, we analyze partial modeling formalisms from the following three aspects:

Consistency. Semantic consistency implies logical consistency but the converse is not true in general: Temporal logic is not expressive enough to detect all forms of inconsistency. In this report, we answer several questions about consistency: Is there a subclass of partial TSs for which semantic and logical consistency coincide? Do TSs outside of this subclass have additional expressive power? Is there a necessary and sufficient condition for ensuring consistency? We show that there is a class of partial TSs for which semantic and logical consistency coincide. We call this class *monotone* because of the monotonicity condition we impose on the transition relation. The class of monotone TSs is as expressive as the class of all partial TSs. Thus, for every partial TS, there is an equivalent monotone one.

At a first glance, it may appear that a structural requirement “every *must* transition is also a *may* transition” is sufficient and necessary to guarantee both semantic and logical consistency. However, this is not the case. We show that for logical consistency, this requirement is sufficient but not necessary: weaker condition exists. For semantic consistency, the requirement is neither necessary nor sufficient. Instead, for monotone TSs, where semantic and logical consistency coincide, we define an alternative structural condition and show that it is both necessary and sufficient to guarantee consistency.

Expressive Power. We show that all three families of partial TSs, KMTSs, MixTSs, and GKMTSs, are equally expressive: for any partial TS M expressed in one formalism, there exists a partial TS M' in the other such that M and M' approximate the same set of concrete systems. That is, neither hyper-transitions nor restrictions on *may* and *must* transitions affect expressiveness. They do, however, affect the size of the models: GKMTSs and KMTSs can be converted to semantically equivalent MixTSs of (possibly exponentially) smaller or equal size. Dams and Namjoshi have shown that the three families of partial TSs are less expressive than tree automata [4]. We complete the picture by showing the expressive equivalence *between* these families.

Model Checking. We call a semantics of temporal logic *inductive* if it is defined inductively on the syntax of the logic. We refer to the typical inductive semantics of L_μ on partial TSs as the *Standard Inductive Semantics* (SIS). This is the semantics most widely used in other works on this subject as well as in practice. A GKMTS G can prove/disprove more properties under SIS than either a MixTS or KMTS obtained from G by semantics-preserving translation. However, while both MixTSs and KMTSs have been used in practical symbolic model checkers (e.g., [8, 2]), the direct use of GKMTSs has been hampered by the difficulty of encoding hyper-transitions into BDDs. To address this problem, we develop a new semantics, called *reduced* (RIS), that is inductive (and tractable) but is more precise than SIS. We show that GKMTSs and MixTSs are equivalent with respect to RIS, and give an efficient symbolic model checking procedure for RIS. The outcome is an algorithm that combines the benefits of the efficient symbolic encoding of MixTSs with the model checking precision of GKMTSs.

To show the practicality of the above result, we develop a symbolic model checking algorithm with respect to RIS and apply it to MixTSs constructed using predicate abstraction. We evaluate our implementation empirically against a SIS-based algorithm.

References

1. G. Bruns and P. Godefroid. Generalized Model Checking: Reasoning about Partial State Spaces. In *Proceedings of the 11th International Conference on Concurrency Theory (CONCUR'00)*, volume 1877 of *LNCS*, pages 168–182, August 2000.
2. M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. Multi-Valued Symbolic Model-Checking. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12(4):1–38, 2003.
3. D. Dams, R. Gerth, and O. Grumberg. Abstract Interpretation of Reactive Systems. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 2(19):253–291, 1997.
4. D. Dams and K. S. Namjoshi. Automata as Abstractions. In *Proceedings of the 6th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'05)*, volume 3385 of *LNCS*, pages 216–232, January 2005.
5. L. de Alfaro, P. Godefroid, and R. Jagadeesan. Three-Valued Abstractions of Games: Uncertainty, but with Precision. In *Proceedings of the 19th IEEE Symposium on Logic in Computer Science (LICS'04)*, pages 170–179, July 2004.
6. A. Gurfinkel and M. Chechik. Why Waste a Perfectly Good Abstraction? In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 212–226, March 2006.
7. A. Gurfinkel, O. Wei, and M. Chechik. Systematic Construction of Abstractions for Model-Checking. In *Proceedings of the 7th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'06)*, volume 3855 of *LNCS*, pages 381–397, January 2006.
8. A. Gurfinkel, O. Wei, and M. Chechik. YASM: A Software Model-Checker for Verification and Refutation. In *Proceedings of the 18th International Conference on Computer-Aided Verification (CAV'06)*, volume 4144 of *LNCS*, pages 170–174, August 2006.
9. M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal Transition Systems: A Foundation for Three-Valued Program Analysis. In *Proceedings of 10th European Symposium on Programming (ESOP)*, volume 2028 of *LNCS*, pages 155–169, April 2001.
10. D. Kozen. Results on the Propositional μ -calculus. *Theoretical Computer Science*, 27:334–354, 1983.
11. K. G. Larsen and B. Thomsen. A Modal Process Logic. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science (LICS '88)*, pages 203–210, July 1988.
12. P. Larsen. The Expressive Power of Implicit Specifications. In *Proceedings of the 18th International Colloquium on Automata, Languages and Programming (ICALP'91)*, volume 510 of *LNCS*, pages 204–216, July 1991.
13. S. Shoham and O. Grumberg. Monotonic Abstraction-Refinement for CTL. In *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *LNCS*, pages 546–560, March 2004.
14. O. Wei, A. Gurfinkel, and M. Chechik. Mixed Transition Systems Revisited. In *Proceedings of the 10th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'09)*, volume 5403 of *LNCS*, pages 349–365, January 2009.
15. O. Wei, A. Gurfinkel, and M. Chechik. On the Consistency, Expressiveness, and Precision of Partial Modeling Formalisms. *Information and Computation*, 209(1):20–47, 2011.