Parameterized Verification of Systems with Precise $(0, 1)$-Counter Abstraction

Paul Eichler[1], Teymur Ismikhanov[1], Swen Jacobs[1], and Chana Weil-Kennedy[2]

[1] CISPA Helmholtz Center for Information Security, Germany
[2] IMDEA Software Institute, Spain

Keywords: Parameterized Verification, Finite Abstraction

1 Extended Abstract

Concurrent systems often consist of an arbitrary number of uniform user processes running in parallel, possibly with a distinguished controller process. Given a description of the user and controller protocols and a desired property, the *parameterized model checking problem* (PMCP) asks whether the property holds in the system, regardless of the number of user processes. The PMCP is well-known to be undecidable in general, but a long line of research has valiantly strived for the identification of decidable fragments that support interesting models and properties [20,15,1,17,13,16,2,8].

While many works in this research direction share certain techniques, systems with different communication primitives have usually been studied separately, and it is hard to keep an overview of which problems are decidable for which class of systems, and why. In this work, we show that there is a range of systems, previously studied using different techniques, that can be unified in a single framework. Our framework not only gives a surprisingly simple explanation of existing decidability results for these systems, but also extends both the class of systems and the types of properties that can be verified, and in some cases proves lower complexity bounds for these problems than were previously known.

The main condition of our framework resembles that of well-structured transition systems (WSTS), i.e., compatibility of transitions with a well-quasi order. However, we do not make use of any WSTS techniques, but instead show that a specific finitary abstraction is precise for all systems satisfying the condition. This (0,1)-abstraction is not only fixed for all systems under consideration, but may also be much more concise than the abstraction we would obtain by using the WSTS framework.

Parameterized Systems with precise $(0, 1)$-Abstraction. The systems we consider are based on one control process and an arbitrary number of identical user processes. Processes change state synchronously according to a step relation, usually based on local transitions that may be synchronized based on transition labels. Our framework supports the following communication primitives from the literature:

- *Lossy broadcast*: in a global step, one process takes a broadcast transition $q_1 \xrightarrow{!a} q_1'$, and every process that has a receive transition of the form $q_2 \xrightarrow{?a} q_2'$ may or may not take that transition at the same time.
  Systems based on lossy broadcast [12] are equivalent to the widely studied system model of *reconfigurable broadcast networks* (RBN) [11,7,4,5].
- *Disjunctive guards*: every global step corresponds to a single transition $q_1 \xrightarrow{G_\exists} q_1'$ of one process, which can only be taken if there is another process in one of the states in $G_\exists$ (i.e., transition labels are sets of local states).
  Systems based on *disjunctive guards*, or short: *disjunctive systems*, have been studied extensively, providing cutoff results for different parameterized verification problems [13,14,3,21] as well as some reductions that work without a cutoff [2,22]. Moreover, this model is equivalent to immediate observation (IO) protocols, a subclass of population protocols [19].
- *Synchronization*: in a step of the system, one process takes a transition $q_1 \xrightarrow{a} q_1'$, and every process that has a transition $q_2 \xrightarrow{a} q_2'$ (i.e., with the same label) takes this transition at the same time. Processes that do not have such a transition stay in their current state.
  Synchronization protocols are studied for example in [6,10], where the problem considered is population control.
- *Shared finite-domain variables*: the controller keeps track of the values of shared variables $x_1, \ldots, x_k$ with finite domain $D_1, \ldots, D_k$, respectively. Transitions of the user processes can read the current value of some x_i, or they can update the value of some x_i to a specific value in D_i, forcing the controller to synchronize and take a transition that reflects this change.
  Shared finite-domain protocols, when considered with only one finite domain, are akin to asynchronous shared-memory systems (ASMS) [18], sometimes called register protocols [9].

While some of the works mentioned above use some form of $(0, 1)$-abstraction either explicitly or implicitly, none of them provide a general argument for when this abstraction is sufficient, and it is hard to keep track of the different results and their explanations. In this work, we provide a general condition under which $(0, 1)$-abstraction is correct, thereby providing a surprisingly simple framework that systematizes and explains many of the existing results for the types of systems mentioned above. Moreover, we easily obtain new decidability results for types of systems that have, to the best of our knowledge, not been considered in the literature.

Contributions. We introduce a common framework for the verification of a class of parameterized systems that are well-structured with respect to a specific well-quasi order.

First, we prove that for all such systems, $(0, 1)$-counter abstraction is sound and complete for safety properties, and that lossy broadcast protocols, disjunctive systems, synchronization protocols, and shared finite-domain protocols fall into this class, as well as systems with combinations of these primitives.

Then, we consider the cardinality reachability problem, which subsumes classical parameterized problems like coverability and target, and show that the problem is PSPACE-complete for our class of systems. In particular, the hardness proof introduces a novel type of synchronization primitive that falls into our class.

Moreover, we show that under modest additional assumptions the complexity of these problems is significantly lower.

Finally, we show that properties over finite traces of a fixed number of processes in the parameterized system can also be decided based on the 01-counter system, and slightly improve known results on properties over infinite traces for disjunctive systems.

Our results are surprising in the simplicity of the proof techniques, in particular since many of the models that fall into our class have already been studied extensively. They provide a novel and systematic framework for analyzing a range of different system models that have been studied in isolation so far, and also give rise to new system models that naturally describe behaviors of concurrent systems, but have not been considered in the parameterized verification literature before.

References

1. Abdulla, P.A., Cerans, K., Jonsson, B., Tsay, Y.K.: General decidability theorems for infinite-state systems. In: Proceedings 11th Annual IEEE Symposium on Logic in Computer Science. pp. 313–321. IEEE (1996)
2. Aminof, B., Kotek, T., Rubin, S., Spegni, F., Veith, H.: Parameterized model checking of rendezvous systems. Distributed Comput. **31**(3), 187–222 (2018)
3. Außerlechner, S., Jacobs, S., Khalimov, A.: Tight cutoffs for guarded protocols with fairness. In: VMCAI. Lecture Notes in Computer Science, vol. 9583, pp. 476–494. Springer (2016)
4. Balasubramanian, A.R., Bertrand, N., Markey, N.: Parameterized verification of synchronization in constrained reconfigurable broadcast networks. In: TACAS (2). Lecture Notes in Computer Science, vol. 10806, pp. 38–54. Springer (2018)
5. Balasubramanian, A.R., Guillou, L., Weil-Kennedy, C.: Parameterized analysis of reconfigurable broadcast networks. In: FoSSaCS. Lecture Notes in Computer Science, vol. 13242, pp. 61–80. Springer (2022)
6. Bertrand, N., Dewaskar, M., Genest, B., Gimbert, H., Godbole, A.A.: Controlling a population. Log. Methods Comput. Sci. **15**(3) (2019). https://doi.org/10.23638/LMCS-15(3:6)2019, https://doi.org/10.23638/LMCS-15(3:6)2019
7. Bertrand, N., Fournier, P., Sangnier, A.: Playing with probabilities in reconfigurable broadcast networks. In: FoSSaCS. Lecture Notes in Computer Science, vol. 8412, pp. 134–148. Springer (2014)
8. Bloem, R., Jacobs, S., Khalimov, A., Konnov, I., Rubin, S., Veith, H., Widder, J.: Decidability of Parameterized Verification. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers (2015). https://doi.org/10.2200/S00658ED1V01Y201508DCT013, https://doi.org/10.2200/S00658ED1V01Y201508DCT013

9. Bouyer, P., Markey, N., Randour, M., Sangnier, A., Stan, D.: Reachability in networks of register protocols under stochastic schedulers. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy. LIPIcs, vol. 55, pp. 106:1–106:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016). https://doi.org/10.4230/LIPIcs.ICALP.2016.106, https://doi.org/10.4230/LIPIcs.ICALP.2016.106
10. Colcombet, T., Fijalkow, N., Ohlmann, P.: Controlling a random population. Log. Methods Comput. Sci. **17**(4) (2021). https://doi.org/10.46298/lmcs-17(4:12)2021, https://doi.org/10.46298/lmcs-17(4:12)2021
11. Delzanno, G., Sangnier, A., Traverso, R., Zavattaro, G.: On the complexity of parameterized reachability in reconfigurable broadcast networks. In: FSTTCS. LIPIcs, vol. 18, pp. 289–300. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012)
12. Delzanno, G., Sangnier, A., Zavattaro, G.: Verification of ad hoc networks with node and communication failures. In: FMOODS/FORTE. Lecture Notes in Computer Science, vol. 7273, pp. 235–250. Springer (2012)
13. Emerson, E.A., Kahlon, V.: Reducing model checking of the many to the few. In: McAllester, D.A. (ed.) Automated Deduction - CADE-17, 17th International Conference on Automated Deduction, Pittsburgh, PA, USA, June 17-20, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1831, pp. 236–254. Springer (2000). https://doi.org/10.1007/10721959_19, https://doi.org/10.1007/10721959_19
14. Emerson, E.A., Kahlon, V.: Model checking guarded protocols. In: LICS. pp. 361–370. IEEE Computer Society (2003)
15. Emerson, E.A., Namjoshi, K.S.: Reasoning about rings. In: Cytron, R.K., Lee, P. (eds.) Conference Record of POPL'95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995. pp. 85–94. ACM Press (1995). https://doi.org/10.1145/199448.199468, https://doi.org/10.1145/199448.199468
16. Esparza, J.: Keeping a crowd safe: On the complexity of parameterized verification (invited talk). In: Mayr, E.W., Portier, N. (eds.) 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France. LIPIcs, vol. 25, pp. 1–10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2014). https://doi.org/10.4230/LIPIcs.STACS.2014.1, https://doi.org/10.4230/LIPIcs.STACS.2014.1
17. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: LICS. pp. 352–359. IEEE Computer Society (1999)
18. Esparza, J., Ganty, P., Majumdar, R.: Parameterized verification of asynchronous shared-memory systems. J. ACM **63**(1), 10:1–10:48 (2016). https://doi.org/10.1145/2842603, https://doi.org/10.1145/2842603
19. Esparza, J., Raskin, M.A., Weil-Kennedy, C.: Parameterized analysis of immediate observation petri nets. In: Donatelli, S., Haar, S. (eds.) Application and Theory of Petri Nets and Concurrency - 40th International Conference, PETRI NETS 2019, Aachen, Germany, June 23-28, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11522, pp. 365–385. Springer (2019). https://doi.org/10.1007/978-3-030-21571-2_20, https://doi.org/10.1007/978-3-030-21571-2_20
20. German, S.M., Sistla, A.P.: Reasoning about systems with many processes. J. ACM **39**(3), 675–735 (1992)

21. Jacobs, S., Sakr, M.: Analyzing guarded protocols: Better cutoffs, more systems, more expressivity. In: VMCAI. Lecture Notes in Computer Science, vol. 10747, pp. 247–268. Springer (2018)
22. Jacobs, S., Sakr, M., Völp, M.: Automatic repair and deadlock detection for parameterized systems. In: FMCAD. pp. 225–234. IEEE (2022)