Enhancing Property-Directed Reachability for Chemical Reaction Networks

Landon Taylor
Electrical and Computer Engineering
Utah State University
Logan, Utah, USA
0000-0002-4071-3625

Zhen Zhang
Electrical and Computer Engineering
Utah State University
Logan, Utah, USA
0000-0002-8269-9489

***Abstract*—Property-Directed Reachability (PDR) is a scalable method for inductively checking reachability of system models. PDR is specialized for Boolean and integer systems, but it is currently unable to provide probability information for continuous-time systems like Chemical Reaction Networks (CRNs). This research advances PDR to bound variables in infinite-state CRNs and to estimate a target state's reachability probability.**

***Index Terms*—Property-Directed Reachability, Chemical Reaction Networks, Probabilistic Verification**

I. Background

Property-Directed Reachability (PDR), also known as IC3, is a highly-scalable symbolic verification technique designed for reachability analysis of Boolean systems [1], [2]. PDR has since been expanded and revised to verify systems with quantifier free formulae over bitvectors [3] and Markov decision processes [4]. Its scalability has increased its popularity, and efforts have recently been made to improve its efficiency and broaden its use cases [4]–[12].

In a model with initial state I and safety property P, PDR is highly effective at either generating a single trace from I to a *target state* satisfying $\neg P$ or providing an invariant as a proof that P holds. PDR analyzes a model using relatively inductive frames. A frame F_k contains an *overapproximation* of states reachable from I within k steps. In other words, F_k contains reachable regions (collections of states) from F_{k-1} in one step. In traditional PDR, execution terminates when the whole reachable state space satisfies P or when PDR cannot exclude a target state from the reachable area. **This work leverages PDR's ability to find traces to $\neg P$ to obtain the probability of a CRN behavior of interest.**

A simple example is shown in Figure 1. Frame F_0 indicates that only I is reachable in zero steps. In F_1, unreachable regions from F_0 are blocked (shown in gray). F_1 proves $\neg P$ is unreachable from F_0 within one step. Then frame F_2 is created, and unreachable regions from F_1 are blocked. After two steps, more states are reachable, but $\neg P$ is still blocked. After creating F_3, it is impossible to block every target state. Thus, a trace is found inductively as follows: state s_2 in F_2 can reach $\neg P$ in F_3 in one step. State s_1 in F_1 can reach s_2, and I can reach s_1. Thus, the trace marked by the red dotted arrows is a counterexample trace from I to $\neg P$ and the safety property P does not hold in this model.

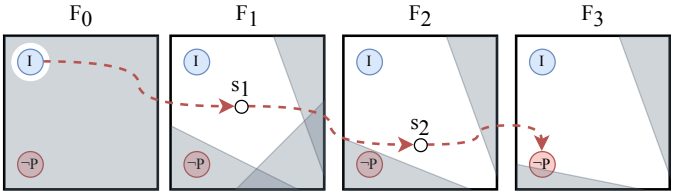

Fig. 1. Single 3-step PDR Counterexample

II. Project Goals

While single counterexample generation makes PDR a useful tool for reachability analysis in a variety of systems [1], [3], [13], PDR can be adapted and specialized to Chemical Reaction Networks (CRNs) modeled as Continuous-Time Markov Chains (CTMCs). CRNs are a formal language for modeling chemical kinetics in synthetic biological systems. CRNs often include rare events that challenge state-of-the-art Probabilistic Model Checking (PMC) tools like PRISM [14], Storm [15], and STAMINA [16]. These challenges are demonstrated in [17]. While probabilistic extensions of PDR are promising [4], [5], these tools are designed for *discrete-time* Markov models. Further, these tools verify a *user-provided* upper bound on the probability of failure, requiring either advance knowledge of the model or guess-and-check probability upper-bound calculations. These current drawbacks bar current PDR expansions from being used in CRN evaluation. This project tackles three main challenges for CRN analysis and verification:

1) Obtaining an accurate rare-event probability for CRNs requires the enumeration of *many* traces from I to $\neg P$;
2) Finding probabilities for a CRN requires CTMC numerical analysis and explicit state space construction; and
3) Bounding variables in infinite-state CRNs requires user knowledge, so users often have to guess bounds that lead to state explosion or remove important states.

We propose the following two solutions to these problems:

1) We present a *fully-automated* method to bound a state space by gathering information from PDR frames. These bounds can allow existing PMC tools to verify an explicit state space that includes relevant traces; and

2) We propose an alternative to explicit state space enumeration: a layered expansion for PDR that symbolically estimates the probability of reaching $\neg P$ in a CRN.

III. PDR-Based Variable Bounding

We propose a CRN-optimized version of PDR to automatically bound variables in an otherwise infinite state space. These bounds enable efficient PMC and resolve state explosion. We propose three adaptations to PDR to achieve this:

1) When blocking regions, use information from the CRN model to suggest smart bounds for each variable. For example, if a variable x can increase by up to two at each step, a reasonable bound at step k is $x \leq 2k$. PDR can then create bounds for each variable in the model.
2) Once one trace from I to $\neg P$ is found in n steps, frame F_n contains every trace of n steps. CRNs are highly concurrent, so allowing an additional few steps can significantly increase the probability of a CRN reaching $\neg P$ [17]. PDR can thus continue to propagate blocked regions into k additional frames. Variable bounds are then derived from blocked regions at frame F_{n+k}, enabling a PMC tool to find all traces of up to $n+k$ steps.
3) Bounds can be further tightened by evaluating backward reachability. Consider the following CRN modification:
   - Consider $\neg I$ as the safety property
   - Consider $\neg P$ as the initial state formula
   - Reverse all transitions

   By evaluating this modified model, PDR obtains a backward-reachable region. That is, it finds all states that can reach a target state within n steps, regardless of the initial state. Consider the reachable states from standard PDR Region A and the backward-reachable states Region B. The region $A \cap B$ thus contains all the traces that start at I *and* end at $\neg P$, as shown in Figure 2. This produces a tighter bound on the state space and can improve PMC efficiency.

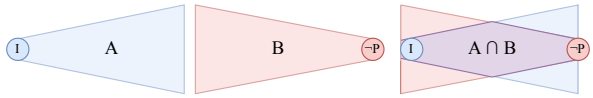

Fig. 2. Forward vs. Backward Reachability

Enumerating many traces explicitly is challenging in PDR, but PMC tools like PRISM and Storm are efficient at finding probabilities given a well-bounded state space. The probability returned by PMC using these bounds is a lower bound on the true probability because the model is only guaranteed to include traces up to a finite number of steps. As traces accumulate, the sum of the probability of those traces asymptotically approaches the true probability. Thus, in a practical case, this method likely allows PMC to find the true probability.

IV. Using PDR to Estimate Probability

We propose a layered approach to PDR that enables incremental probability estimation. This approach involves three layers, described as follows and shown in Figure 3:

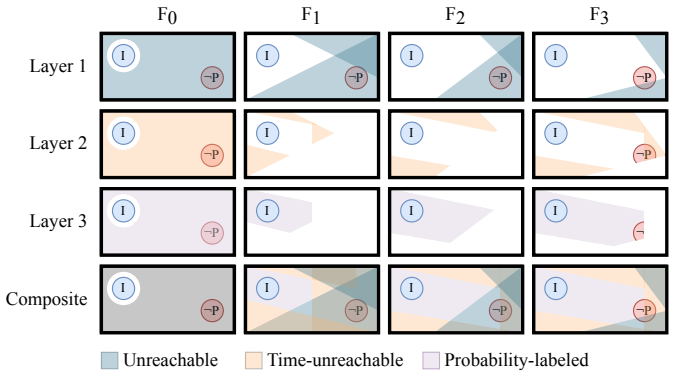

Fig. 3. Layered PDR Example

Layer 1: Reachability. Layer 1 is described in Section III. This layer is probability-agnostic and blocks unreachable states in each frame. The unblocked/white area in this layer is an overapproximation of the area reachable within k steps.

Layer 2: Time-Bounding. Layer 2 analyzes CRN time-bounded reachability information. CRN properties are constrained by a time bound that can be used to decide what regions of a state space are worth exploring. This layer groups the reachable area of Layer 1 into regions that allow for a calculation of bounds for each region's mean dwell time (i.e., the average amount of time a CRN expects to spend in a region). Using PDR's inductive checking, this layer accumulates mean dwell times and blocks regions that take too long to reach a target state. This layer blocks a larger region of the state space than reachability analysis alone, but it causes this method to give an estimate of the final probability, not a guarantee. The white/unblocked region in this layer is an overapproximation of the region that, on average, can reach a target state within the CRN property time bound.

Layer 3: Probability-Labeled Regions This layer is the intersection of the reachable regions at each frame in Layers 1 and 2. This layer groups states into regions of states with similar properties and labels each region with a probability. The probability labels in each region at each frame are the upper and lower bounds on the probability of reaching the region from the previous frame. The final probability upper and lower bounds can then be computed inductively starting at regions satisfying $\neg P$ in the final frame.

V. Conclusion

This project aims to advance PDR for probabilistic CRN verification. Providing explicit bounds for a CRN state space enables more efficient verification while removing user guesswork. When cutting-edge PMC tools cannot compute a probability, a layered expansion of PDR enables the inductive estimation of probability bounds.

REFERENCES

[1] A. R. Bradley, "SAT-Based Model Checking without Unrolling," in *Verification, Model Checking, and Abstract Interpretation*, ser. Lecture Notes in Computer Science, R. Jhala and D. Schmidt, Eds. Springer, 2011, pp. 70–87.

[2] N. Een, A. Mishchenko, and R. Brayton, "Efficient implementation of property directed reachability," in *Proceedings of the International Conference on Formal Methods in Computer-Aided Design*, ser. FMCAD '11. Austin, Texas: FMCAD Inc, Oct. 2011, pp. 125–134.

[3] T. Welp and A. Kuehlmann, "QF_BV Model Checking with Property Directed Reachability," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*. IEEE Conference Publications, 2013, pp. 791–796.

[4] K. Batz, S. Junges, B. L. Kaminski, J.-P. Katoen, C. Matheja, and P. Schröer, "PrIC3: Property Directed Reachability for MDPs," in *Computer Aided Verification*, S. K. Lahiri and C. Wang, Eds., vol. 12225. Springer International Publishing, 2020, pp. 512–538.

[5] E. Polgreen, M. Brain, M. Fraenzle, and A. Abate, "Verifying Reachability Properties in Markov Chains via Incremental Induction," Sep. 2019.

[6] A. Gurfinkel, A. Ivrii, and Y. Vizel, "IC3 with Internal Signals," in *2021 Formal Methods in Computer Aided Design (FMCAD)*, 2021, p. 9.

[7] A. Gurfinkel and A. Ivrii, "K-induction without unrolling," in *2017 Formal Methods in Computer Aided Design (FMCAD)*, Oct. 2017, pp. 148–155.

[8] R. Dureja and K. Y. Rozier, "FuseIC3: An algorithm for checking large design spaces," in *2017 Formal Methods in Computer Aided Design (FMCAD)*. IEEE, Oct. 2017, pp. 164–171.

[9] M. Blicha, A. E. J. Hyvärinen, M. Marescotti, and N. Sharygina, "A Cooperative Parallelization Approach for Property-Directed k-Induction," in *Verification, Model Checking, and Abstract Interpretation*, ser. Lecture Notes in Computer Science, D. Beyer and D. Zufferey, Eds. Springer International Publishing, 2020, pp. 270–292.

[10] S. Chaki and D. Karimi, "Model Checking with Multi-threaded IC3 Portfolios," in *Verification, Model Checking, and Abstract Interpretation*, ser. Lecture Notes in Computer Science, B. Jobstmann and K. R. M. Leino, Eds. Springer, 2016, pp. 517–535.

[11] D. Jovanović and B. Dutertre, "Property-directed k-induction," in *2016 Formal Methods in Computer-Aided Design (FMCAD)*, Oct. 2016, pp. 85–92.

[12] M. Marescotti, A. Gurfinkel, A. E. J. Hyvärinen, and N. Sharygina, "Designing parallel PDR," in *2017 Formal Methods in Computer Aided Design (FMCAD)*, Oct. 2017, pp. 156–163.

[13] K. Hoder and N. Bjørner, "Generalized Property Directed Reachability," in *Theory and Applications of Satisfiability Testing – SAT 2012*, ser. Lecture Notes in Computer Science, A. Cimatti and R. Sebastiani, Eds. Berlin, Heidelberg: Springer, 2012, pp. 157–171.

[14] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Proceedings of the 23rd International Conference on Computer Aided Verification*, ser. CAV'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 585–591.

[15] C. Hensel, S. Junges, J.-P. Katoen, T. Quatmann, and M. Volk, "The probabilistic model checker Storm," *International Journal on Software Tools for Technology Transfer*, vol. 24, no. 4, pp. 589–610, Aug. 2022.

[16] R. Roberts, T. Neupane, L. Buecherl, C. J. Myers, and Z. Zhang, "STAMINA 2.0: Improving scalability of infinite-state stochastic model checking," in *Verification, Model Checking, and Abstract Interpretation*, B. Finkbeiner and T. Wies, Eds. Cham: Springer International Publishing, 2022, pp. 319–331.

[17] B. Israelsen, L. Taylor, and Z. Zhang, "Efficient trace generation for rare-event analysis in chemical reaction networks," in *Model Checking Software*, G. Caltais and C. Schilling, Eds. Cham: Springer Nature Switzerland, 2023, pp. 83–102.