fbPDR: In-depth combination of forward and backward analysis in Property Directed Reachability (extended abstract)

Tobias Seufert and Christoph Scholl
University of Freiburg, Freiburg, Germany, {seufert,scholl}@informatik.uni-freiburg.de

Abstract

We describe a thoroughly interweaved forward and backward version of PDR/IC3 called *fbPDR*. Motivated by the complementary strengths of PDR and Reverse PDR, *fbPDR* enables beneficial collaboration between the two and lifts the combination to a new level. We lay the theoretical foundations for sharing information between PDR and Reverse PDR and demonstrate the effectiveness of our approach on benchmarks from the Hardware Model Checking Competition.

1 Introduction

Nowadays PDR (or IC3) [1] [2] is considered as one of the most powerful methods in hardware verification. Unlike others it does not unroll a transition relation. It incrementally strengthens a proof until a safe invariant or a counterexample is found. PDR in its usual definition has a 'fixed direction': It considers overapproximations of state sets reachable from the *initial states* in k or less steps.
This work is motivated by observations already made in [3] showing that a combination of (forward) PDR and (backward) Reverse PDR is worthwile. Reverse PDR computes over-approximations of state sets which can reach *unsafe states* in k or less steps. In [4] we examine Reverse PDR thoroughly and enable communication between PDR and Reverse PDR via proof obligations.
In our most recent work we lift the combination of PDR and Reverse PDR to a next level [5]. Our algorithm really intertwines PDR and Reverse PDR reasoning by strengthening one trace using blocked cubes learnt from the other. We show that both communication via proof obligations and strengthening one trace with information from the other can indeed be used successfully in combination.
Here we describe our forward and backward version of PDR/IC3 called *fbPDR*. Both PDR and Reverse PDR profit from all the information gathered by its counterpart and we observe a significant speedup in both finding counterexamples as well as safe inductive invariants.

2 Communication via Proof Obligations

One kind of information exchange between the two directions of PDR takes place on the basis of proof obligations. Sets of proof obligations of the original forward PDR represent underapproximations of the set of states from which the unsafe states (or *unsafe*) can be reached.
The reason for this is that a proof obligation (s,k,d) in time frame k is a state s for which it has been shown that there is a path of length d from s to *unsafe*. Therefore PDR is obliged to prove that s is unreachable within $\leq k$ steps from the initial states (or *init*). Hence, in Reverse PDR the set of unsafe states can be extended by the proof obligations from forward PDR as a "target enlargement". Extending *unsafe* in Reverse PDR has two effects:
1.) During Reverse PDR, the intersection of a cube s that can be reached from *init* (i.e., of a proof obligation in Reverse PDR) with the extended *unsafe* may now be non-empty, because of a non-empty intersection of s with a proof obligation from forward PDR. Due to this non-empty intersection, a counterexample, i.e., a trace from *init* to *unsafe*, has been found where the first part of the trace (reaching s from *init*) has been constructed by Reverse PDR and the second part (reaching *unsafe* from s) has been constructed by forward PDR.
2.) Sometimes in (Reverse) PDR the generalization of blocked cubes (learnt clauses) s into $\hat{s}_1$ for unreachable proof obligations (technically, this corresponds to unsatisfiable SAT solver calls that allow for literal dropping, see e.g. [2]) may be unnecessarily large such that it prevents early convergence of the procedure. In the combined algorithm unnecessary large generalizations are restricted by the stronger requirement that a generalized cube $\hat{s}_2$ does not intersect the extended unsafe states, not only *unsafe* as in the original Reverse PDR. If a larger $\hat{s}_1$ that contains states from which *unsafe* can be reached (as has been proved by forward PDR) would be removed by learning clause $\neg\hat{s}_1$, this clause could not be part of any safe inductive invariant (since it excludes states which reach unsafe eventually).

Of course, a dual argumentation is possible for transferring information from Reverse PDR to PDR: Sets of proof obligations of Reverse PDR represent underapproximations of the set of states which can be reached from *init*. So in the original PDR the set of initial states can be extended by these proof obligations.

3 Learning new Lemmas from Reverse PDR

Basically, PDR gathers information in terms of proof obligations and learnt clauses (lemmas). In the previous section we presented the capabilities of sharing the proof obli-

gation part of this information. Now we want to focus on *learning new clauses (lemmas)* for PDR from lemmas in Reverse PDR.

Note that in the following our analysis considers only the transfer of information from Reverse PDR to PDR. However, all procedures also apply the other way around considering the characteristics of PDR and Reverse PDR.

Reverse PDR maintains a trace $RR_0, RR_1, \ldots, RR_N$ of clause sets. RR_n represents an overapproximation of states which are able to reach *unsafe* within $0 \leq j \leq n$ steps. PDR maintains a trace $R_0, R_1, \ldots, R_N$ of clause sets where R_n represents an overapproximation of states reachable from *init* within $0 \leq j \leq n$ steps. It holds that all clause sets are supersets of the ones with higher indices, i.e., $R_i \sqsupseteq R_{i+1}$ and $RR_i \sqsupseteq RR_{i+1}$. In contrast, if we consider R_i (RR_i) semantically as the state sets represented by the corresponding clause sets, we have $R_i \subseteq R_{i+1}$ and $RR_i \subseteq RR_{i+1}$.

For a Reverse PDR clause (lemma) $c \in RR_{N-i}$ with $c = \bar{s}$ it is not clear how to make use of this information in PDR where we work with underapproximations of states reaching *unsafe* (i.e., proof obligations) and overapproximations of states reachable from *init* (i.e., lemmas). In contrast, by looking at sets RR_{N-i} as a whole, we can extract useful information:

Theorem 1. *Given a Reverse PDR trace of length N and a PDR trace of length N'. Let s be an arbitrary cube $s \subseteq RR_{N-(i+1)}$ with $0 \leq i \leq N-2$. If we strengthen the PDR trace by blocking s in all frames $1 \leq k \leq \min(i, N')$, i.e. by setting $R_k = R_k \wedge \bar{s}$, then in the resulting PDR trace the state sets R_i with $0 \leq i \leq N$ still overapproximate the sets of states reachable from* init *in $\leq i$ steps. Moreover, the property of syntactical inclusion of clause sets $R_{i+1} \sqsubseteq R_i$ for $1 \leq i \leq N-1$ and semantical inclusion $R_i \subseteq R_{i+1}$ for $0 \leq i \leq N-1$ is preserved by the strengthening.*

A comprehensive proof can be found in [5]. The intuition behind this is that $RR_{N-(i+1)}$ (after discharging all proof-obligations for the trace $RR_0, \ldots, RR_{N-1}$) excludes an overapproximation of the states which are reachable from *init* within $\leq i$ steps. Thus $RR_{N-(i+1)}$ is an underapproximation of all states which are *not* reachable from *init* within $\leq i$ steps. *Excluding* (blocking) such states from R_k with $1 \leq k \leq \min(i, N')$ in forward PDR maintains the property that R_k overapproximates the states reachable from *init* in $\leq k$ steps.

To strengthen a PDR trace according to Thrm. 1 we have to extract subcubes from $RR_{N-(i+1)}$ provided by Reverse PDR. $RR_{N-(i+1)}$ is given as a CNF, thus extracting all subcubes amounts to a CNF to DNF conversion, and extracting a restricted number of good, i.e., short, subcubes means computing only a part of the corresponding DNF. The naive way of CNF to DNF conversion using the law of distributivity can lead to an exponential growth. Another possibility is to negate the CNF, use Plaisted-Greenbaum-Transformation [6] for translating the DNF into CNF, and negate the result again. However, this method may have disadvantages as well: The number of computed cubes is linear in the size of the CNF, but we may be interested in even more condensed information to be transferred. Moreover, we have to introduce new auxiliary variables which act as additional state space variables.

Thus we are using a SAT-based method to pick a small number of preferably short and informative subcubes [5].

4 fbPDR

Our implementation *fbPDR*[1] runs PDR and Reverse PDR in alternation (30s slices) and implements both communication methods presented in Sects. 2 and 3. We also implemented the feature that PDR and Reverse PDR traces with length N and N' can always be extended to length $max(N, N')$ and all proof-obligations which represent counterexamples of length $\leq max(N, N')$ can immediately be discharged.

Experiments of the benchmark set of HWMCC'15 and '17 show promising results comparing the latest version of *fbPDR* to the version only communicating via proof obligations [4] as well as the default configuration of ABC's PDR[2] and ic3ref[3] (see Fig. 1).

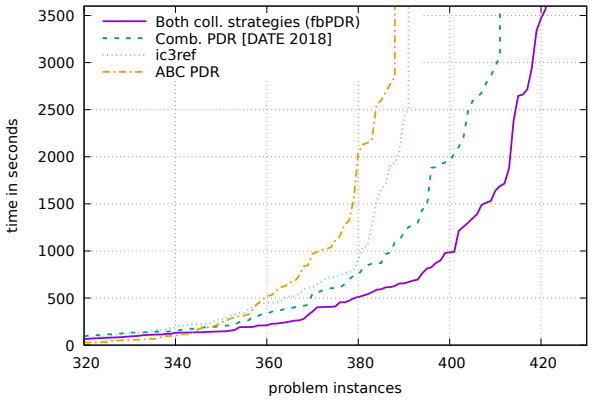

Figure 1 Comparison of implementations

References

[1] A. R. Bradley, "Sat-based model checking without unrolling," in *VMCAI*, 2011, pp. 70–87.

[2] N. Eén, A. Mishchenko, and R. K. Brayton, "Efficient implementation of property directed reachability," in *FMCAD*, 2011, pp. 125–134.

[3] T. Seufert and C. Scholl, "Sequential verification using Reverse PDR," in *MBMV*, 2017, pp. 79–89.

[4] ——, "Combining PDR and reverse PDR for hardware model checking," in *DATE*, 2018, pp. 49–54.

[5] ——, "fbPDR: In-depth combination of forward and backward analysis in property directed reachability," in *to appear in DATE*, 2019.

[6] D. A. Plaisted and S. Greenbaum, "A structure-preserving clause form translation," in *Journal of Symbolic Computation*, 1986, pp. 293–304.

[1] We provide result tables and binaries under `https://www.dropbox.com/s/ckbzq6kd10aebod/fbPDR.zip?dl=0`.

[2] `https://bitbucket.org/alanmi/abc`, downl. on 9/10/2017

[3] `https://github.com/IC3ref`, downl. in Sep. 2016