Intertwined Forward-Backward Reachability Analysis Using Interpolants

Yakir Vizel[1], Orna Grumberg[1], and Sharon Shoham[2]

[1] Computer Science Department, The Technion, Haifa, Israel
[2] School of Computer Science, Academic College of Tel Aviv-Yaffo

Abstract. In this work we develop a novel SAT-based verification approach which is based on interpolation. The novelty of our approach is in extracting interpolants in both forward and backward manner and exploiting them for an *intertwined approximated forward and backward reachability analysis*. Our approach is also mostly *local* and avoids unrolling of the checked model as much as possible. This results in an efficient and complete SAT-based verification algorithm.
We implemented our algorithm and compared it with both McMillan's interpolation-based algorithm and with IC3, on real-life industrial designs as well as on examples from the HWMCC'11 benchmark. In many cases, our algorithm outperformed both methods.

1 Introduction

In this work we develop a novel SAT-based verification approach based on interpolation. The novelty of our approach is in extracting interpolants in both forward and backward manner and exploiting them for an *intertwined approximated forward and backward reachability analysis*. Our approach is also mostly *local* and avoids unrolling of the checked model as much as possible. This results in an efficient and complete SAT-based algorithm.

SAT-based model checking is a highly successful approach for the verification of real-life designs from both hardware and software domains. In its early days SAT-based model checking was used mostly for bug hunting. The introduction of *interpolation* [7] enabled an efficient complete algorithm, referred to as Interpolation-based model checking (ITP) [11].

ITP uses interpolation to extract an over-approximation of a set of reachable states from a proof of unsatisfiability, generated by a SAT-solver. This fact enables to perform a SAT-based reachability analysis. The set of reachable states computed by the reachability analysis is used by ITP to check if a system M satisfies a safety property AGp.

In [1] an alternative SAT-based algorithm, called IC3, is introduced. Similarly to ITP, IC3 also computes over-approximations of sets of reachable states. However, ITP unrolls the model in order to obtain more precise approximations. In many cases, this is a bottleneck of the approach. IC3, on the other hand, improves the precision of the approximations by performing many local checks that do not require unrolling.

Both ITP and IC3 compute over-approximations of the sets of states obtained by a *forward reachability analysis*. The forward analysis starts from the initial states of M, and iteratively computes predecessors while making sure that no bad state violating p is reached. Verification based on reachability can also be performed in a dual manner

using a *backward reachability analysis*. The backward analysis starts from the states satisfying $\neg p$ and iteratively computes ancestors while making sure that no initial state is reached.

Traditionally, BDD-based verification methods [6] use both forward and backward analyses [3,13], while SAT-based methods mainly implement the forward one. Recently, a few works considered backward analysis in the context of SAT as well (e.g. [2,8]). Most such works use forward and backward analyses independently of each other, or use a weak combination of the two, such as replacing the role of the initial states in the backward analysis by the reachable states computed by a forward analysis.

In this work we propose an interpolation-based verification method that applies mostly local checks and avoids unrolling of the model as much as possible. Our approach combines approximated forward and backward analyses in a tight and intertwined way, and uses each of them to enhance the precision of the other. Thus, the tight combination of the two analyses replaces unrolling in enhancing the precision of the computed over-approximated sets of states.

Our work uses the observation that a single SAT check entails information both about states reachable from the initial states (via post-image operations) and about states that reach the bad-states (via pre-image operations). We exemplify this observation by examining the propositional formula $\mathit{INIT}(V) \wedge \mathit{TR}(V, V') \wedge \neg p(V')$ where *INIT* and $\neg p$ describe the sets of initial states and bad states, respectively, and $\mathit{TR}(V, V')$ describes the transition relation. If this formula is satisfiable, then there exists a path of length one from the initial states to the bad states. If it is unsatisfiable, then all states reachable from the initial states in one transition are a subset of p. This fact is often used in forward reachability. We now note that the unsatisfiability of this formula can be used in backward reachability as well. This can be done by interpreting it as "all states that can reach the bad states in one transition are disjoint from the initial states".

We exploit this dual observation by extracting two different interpolants from the unsatisfiabe formula $\mathit{INIT}(V) \wedge \mathit{TR}(V, V') \wedge \neg p(V')$. The *forward interpolant* (the one used in ITP) provides an over-approximation of the post-image of *INIT* which is disjoint from $\neg p$. The *backward interpolant*, computed for the same formula when it is read backward, from right to left, provides an over-approximation of the pre-image of $\neg p$ which is disjoint from *INIT*.

We use the above observation as a key element in traversing the state space in a dual fashion, both forward from the initial states and backwards from the bad states.

Our algorithm, *Dual Approximated Reachability* (DAR), computes a *Forward Reachability Sequence* $\bar{F} = \langle F_0, F_1, \ldots \rangle$, and a *Backward Reachability Sequence* $\bar{B} = \langle B_0, B_1, \ldots \rangle$. The set F_i represents an over-approximation of the set of states which are reachable from *INIT* in exactly i transitions. Further, F_i is disjoint from $\neg p$. Similarly, B_i represents an over-approximation of the set of states that can reach $\neg p$ in exactly i transitions, and it is also disjoint from *INIT*. Thus, the existence of either $\bar{F}$ or $\bar{B}$ of length n ensures that no counterexample of length n exists in M.

The goal of DAR is to gradually strengthen (make more precise) and extend $\bar{F}$ and $\bar{B}$, until a counterexample is found or until one of $\bar{F}$ or $\bar{B}$ reaches a *fixpoint*, that is, no new states are found when the sequence is further extended. To do this, DAR employs local strengthening phases, assisted by a global strengthening phase, when needed. Only

the global strengthening involves unrolling. Thus, the number of unrolling applications is limited. In addition, the depth of the unrolling is also limited.

Initially, $\bar{F} = \langle F_0 \rangle$ and $\bar{B} = \langle B_0 \rangle$, where $F_0 = \mathit{INIT}$ and $B_0 = \neg p$. At iteration n, we define the sequence $\Pi = \langle\ \mathit{INIT},\ F_1 \wedge B_n,\ F_2 \wedge B_{n-1},\ \ldots,\ F_n \wedge B_1,\ \neg p\ \rangle$. Π represents an over-approximation of the set of all possible paths from *INIT* to $\neg p$ of length $n+1$ in M. That is, Π over-approximates the set of all counterexamples in M of length $n+1$. DAR attempts to show that Π represents no counterexample.

The *local strengthening phase* checks whether there are in fact transitions between every two consecutive sets in Π. It turns out that this can be done by applying local checks of the form $F_i(V) \wedge \mathit{TR}(V,V') \wedge B_{n-i}(V')$. If such a formula is unsatisfiable, then no transition exists from $F_i \wedge B_{n-i+1}$ to its successor along Π, thus no counterexample of length $n+1$ exists. This can also be understood by observing that the unsatisfiability of $F_i(V) \wedge \mathit{TR}(V,V') \wedge B_{n-i}(V')$ means that the states reachable from the initial states in i transitions cannot reach B_{n-i} in one transition. Since B_{n-i} includes all states reaching $\neg p$ in $n-i$ transitions, no counterexample of length $n+1$ exists.

In this case, the forward interpolant of $F_i(V) \wedge \mathit{TR}(V,V') \wedge B_{n-i}(V')$ is used to strengthen F_{i+1} while the backward interpolant strengthens B_{n-i+1}. Strengthening is now propagated along $\bar{F}$ and $\bar{B}$. This reflects the fact that the components of one sequence are strengthened based on the components of the other everywhere along the sequences, making the analyses closely intertwined. Next, $\bar{F}$ and $\bar{B}$ are extended by initializing F_{n+1} to be the forward interpolant of $F_n(V) \wedge \mathit{TR}(V,V') \wedge B_0(V')$ and B_{n+1} to be the backward interpolant of $F_0(V) \wedge \mathit{TR}(V,V') \wedge B_n(V')$.

The *global strengthening phase* is applied when $F_i(V) \wedge \mathit{TR}(V,V') \wedge B_{n-i}(V')$ is satisfiable for *all* i. This implies that a transition exists between every two consecutive sets in Π, making local reasoning insufficient. We therefore gradually unroll the model M and check whether the states in $F_i \wedge B_{n-i+1}$ are *unreachable* from *INIT* via i transitions of M. Once we find such an i, the unrolling can stop. We are certain that no counterexample of length $n+1$ exists. We strengthen $\bar{F}$ up to depth i using an interpolation-sequence [10], and return to the local strengthening phase for further strengthening and for extending $\bar{F}$ and $\bar{B}$ to length $n+1$.

We implemented our DAR algorithm and compared it to both ITP and IC3, on real-life industrial designs as well as examples from the HWMCC'11 benchmark. In many cases, our algorithm outperformed both methods. We noticed that the number of iterations where global strengthening was needed, as well as the depth of the unrolling in the global strengthening phase is often smaller relative to the length of $\bar{F}$ and $\bar{B}$. This reflects the fact that our use of unrolling is limited.

To summarize, the novelty of our approach is twofold. It suggests a SAT-based intertwined forward-backward reachability analysis. Further, the reachability analysis is interpolation-based. Yet, it is mostly local and avoids unrolling as much as possible.

1.1 Related Work

Several works use interpolation in the context of model checking. Interpolation-based model checking (ITP) was initially introduced in [11]. Similarly to ITP, DAR also uses interpolation to compute over-approximated sets of reachable states. However, ITP

computes interpolants based on an unrolled formula and increases unrolling to make the over-approximation more precise. DAR, on the other hand, mostly avoids unrolling and uses backward and forward interpolants from local checks for strengthening. In addition, ITP restarts when it finds a spurious counterexample, increasing the depth of unrolling. In contrast, DAR keeps strengthening the computed over-approximations from previous iterations. In [2] improvements for ITP are suggested. They implement a backward-traversal using interpolants. Unlike our method, their backward traversal is an adaptation of ITP and is not tightly integrated with the forward traversal.

The work in [8] is also based on ITP in the sense of computing interpolants based on unrolling of the model, where the depth of unrolling increases in each iteration. Their work integrates the use of forward and backward analyses: in each iteration the result of the backward analysis is used to restrict the initial states and the result of the forward analysis is used to restrict the bad states. Our approach, on the other hand, uses the result of the forward analysis to strengthen *all intermediate sets* of $\bar{B}$. Similarly the result of the backward analysis stregthens $\bar{F}$.

Interpolation-sequence, which extends the notion of an interpolant for a sequence of formulas has been proposed and used for model checking [10,12,14,4]. DAR makes a similar use of interpolation-sequence in its global strengthening phase. In contrast to the other methods, interpolation-sequence is not a key element of DAR since it is only applied occasionally. Further, it is applied to a restricted depth of unrolling.

The introduction of IC3 [1] suggested a different way to compute information about reachable states. Unlike interpolation-based approaches IC3 requires no unrolling and is based on inductive reasoning. The main difference between DAR and IC3 is in the way they strengthen the over-approximated sets of states. IC3 finds a state that can reach $\neg p$ and if it concludes that this state is not reachable, it tries to generalize this fact and removes more than just one state. DAR on the other hand finds an over-approximation of *all* states that can reach $\neg p$, rather than a single state. It then tries to prove that the entire set is unreachable. Also, when DAR fails to strengthen using local reasoning, it applies a limited unrolling in the global phase. On the other hand, IC3 can fall into state enumeration if generalization is not successful.

2 Preliminaries

Let V be a set of boolean variables. For $v \in V$, v' is used to denote the value of v after one time unit. The set of these variables is denoted by V'. In the general case V^i is used to denote the variables in V after i time units (thus, $V^0 = V$). For a formula η over V^i, we denote by $\eta[V^i \leftarrow V^j]$ the formula obtained from η when for each $v \in V$, v^i is replaced with v^j. We use $\eta(V^i)$ or simply $\eta^{\langle i \rangle}$ to denote $\eta[V \leftarrow V^i]$. In particular, $\eta' = \eta[V \leftarrow V']$. We will use $\mathcal{L}(\eta)$ to denote the variables appearing in η. From now on, all formulas we refer to are *propositional formulas*, unless stated otherwise.

Definition 1. *A* finite transition system *is a triple* $M = (V, \text{INIT}, \text{TR})$ *where* V *is a set of boolean variables,* $\text{INIT}(V)$ *is a formula over* V*, describing the initial states, and* $\text{TR}(V, V')$ *is a formula over* V *and the next-state variables* V'*, describing the transition relation.*

An assignment s assigning values from $\{0,1\}$ to V defines a *state* in M. A formula over V represents a set of states which consists of all the satisfying assignments of the formula. We refer to a formula η over V as a set of states and therefore use the notation $s \in \eta$ for states represented by η. Similarly, a formula η over V, V' represents a set of pairs of states, and we write $(s, s') \in \eta$ for pairs in the set.

A *path* of length n in M is a sequence of states $\pi = s_0, \ldots, s_n$ s.t. $s_0 \in \mathit{INIT}$ and for all $0 \leq i < n$, $(s_i, s_{i+1}) \in \mathit{TR}$. Let AGp be a safety property, where p is a formula over V. A path $\pi = s_0, \ldots, s_n$ in M is a *counterexample* of length n for AGp if $s_n \models \neg p$.

Let Q be a formula over V. The *post-image* of Q w.r.t. M is the set of all states reachable from Q in one transition, defined by the formula $\exists V[Q(V) \wedge \mathit{TR}(V, V')]$ (note that this formula is defined over V'). The *pre-image* of Q w.r.t. M is the set of all states that can reach Q in one transition, defined by $\exists V'[\mathit{TR}(V, V') \wedge Q(V')]$.

Definition 2. *Let M be a transition system and φ and ψ formulas over V. The formula $\Gamma_M(\varphi, \psi) = \varphi(V) \wedge \mathrm{TR}(V, V') \wedge \psi(V')$ is a* local reachability check *w.r.t. M, φ, ψ.*

Whenever M is clear from the context we omit M and write $\Gamma(\varphi, \psi)$.

Let (ϕ^-, ϕ^+) be a pair of formulas. If $\phi^- \wedge \phi^+$ is unsatisfiable, then by [7] we know that there exists an interpolant, defined as follows.

Definition 3 (Interpolant). *Let $\phi^- \wedge \phi^+ \equiv \bot$ be an unsatisfiable formula. An* interpolant *for $\phi^- \wedge \phi^+$, denoted $I(\phi^-, \phi^+)$, is a formula I s.t. (i) $\phi^- \Rightarrow I$, (ii) $I \wedge \phi^+ \equiv \bot$, and (iii) $\mathcal{L}(I) \subseteq \mathcal{L}(\phi^-) \cap \mathcal{L}(\phi^+)$.*

A similar property holds for conjunctions of more than 2 formulas [10,14]:

Definition 4 (Interpolation-Sequence). *Let $\langle A_1, \ldots, A_n \rangle$ be a sequence of formulas s.t. $\bigwedge_{i=1}^{n} A_i \equiv \bot$. An* interpolation-sequence *for $\langle A_1, \ldots, A_n \rangle$ is a sequence $\langle I_0, I_1, \ldots, I_n \rangle$ of formulas s.t.: (i) $I_0 \equiv \top$ and $I_n \equiv \bot$, (ii) For every $0 \leq j < n$, $I_j \wedge A_{j+1} \Rightarrow I_{j+1}$, and (iii) For every $0 < j < n$, $\mathcal{L}(I_j) \subseteq \mathcal{L}(A_1, \ldots, A_j) \cap \mathcal{L}(A_{j+1}, \ldots, A_n)$.*

3 Using Interpolants for Forward and Backward Analysis

3.1 Forward and Backward Interpolants

Interpolation is typically used in model checking in order to compute over-approximated sets of reachable states [11,12,14].

Let R and Q be propositional formulas over V representing sets of states, and let $\mathit{TR}(V, V')$ be a transition relation. Suppose we would like to know if the post image of R is disjoint from Q. This property can be checked by checking the formula $\Gamma(R, Q) = R(V) \wedge \mathit{TR}(V, V') \wedge Q(V')$ for unsatisfiability. If the formula is unsatisfiable then the answer is yes, meaning that Q is not reachable from R in one step. Moreover, consider $\phi^- = R(V) \wedge \mathit{TR}(V, V')$ and $\phi^+ = Q(V')$. An interpolant $I = I(\phi^-, \phi^+)$ satisfies $R(V) \wedge \mathit{TR}(V, V') \Rightarrow I(V')$ and $I(V') \wedge Q(V') \equiv \bot$. Therefore, I represents an over approximation of the post-image of R, and it is also disjoint from Q.

The unsatisfiability of the formula $\Gamma(R,Q) = R(V) \wedge \mathit{TR}(V,V') \wedge Q(V')$ can also be interpreted in a different manner, shedding light on the pre-image of Q. More precisely, the unsatisfiability of the formula states that the pre-image of Q is disjoint from R. This view leads to a different way of using interpolation in this setting. For the backward interpretation, we now define $\phi^- = \mathit{TR}(V,V') \wedge Q(V')$ and $\phi^+ = R(V)$. Again, since $\phi^- \wedge \phi^+$ is unsatisfiable, an interpolant I exists. Formally $\mathit{TR}(V,V') \wedge Q(V') \Rightarrow I(V)$, therefore I is an over-approximation of the pre-image of Q. Moreover, $I \wedge R$ is unsatisfiable and therefore I is disjoint from R.

We conclude that interpolation gives us a way to approximate both post-image and pre-image computations. Formally, we define forward and backward interpolants:

Definition 5 (Forward and Backward Interpolants). *Let R and Q be propositional formulas over V s.t. $\Gamma(R,Q) \equiv \bot$. The* forward interpolant *of $\Gamma(R,Q)$, denoted $\mathit{FI}(R,Q)$, is $I(R(V) \wedge \mathrm{TR}(V,V'), Q(V'))[V' \leftarrow V]$. The* backward interpolant *of $\Gamma(R,Q)$, denoted $\mathit{BI}(R,Q)$, is $I(\mathrm{TR}(V,V') \wedge Q(V'), R(V))$.*

Note that $I(R(V) \wedge \mathit{TR}(V,V'), Q(V'))$ is defined over V'. Therefore, we substitute V' for V in the definition of a forward interpolant. As explained above:

Lemma 1. *$\mathit{FI}(R,Q)$ over-approximates the post-image of R, and is disjoint from Q. Similarly, $\mathit{BI}(R,Q)$ over-approximates the pre-image of Q, and is disjoint from R.*

3.2 Forward and Backward Reachability Sequences

Our model checking algorithm for safety properties, described in Sec. 4, uses forward and backward interpolants for the computation of over-approximated sets of forward and backward reachable states. Technically, we consider both forward and backward reachability approximations:

Definition 6. *A* Forward Reachability Sequence (FRS) *of length n w.r.t. M and a property AGp is a sequence $\bar{F}_{[n]} = \langle F_0, F_1, \ldots, F_n \rangle$ of sets of states s.t.*

- $F_0 = \mathrm{INIT}$
- $F_i(V) \wedge \mathrm{TR}(V,V') \Rightarrow F_{i+1}(V')$ *for* $0 \leq i < n$
- $F_i \Rightarrow p$ *for* $0 \leq i \leq n$.

Definition 7. *A* Backward Reachability Sequence (BRS) *of length n w.r.t. M and a property AGp is a sequence $\bar{B}_{[n]} = \langle B_0, B_1, \ldots, B_n \rangle$ of sets of states s.t.*

- $B_0 = \neg p$.
- $B_{i+1}(V) \Leftarrow \mathrm{TR}(V,V') \wedge B_i(V')$ *for* $0 < i \leq n$.
- $B_i \Rightarrow \neg\mathrm{INIT}$ *for* $0 \leq i \leq n$.

When n is clear from the context, we simply use $\bar{F}$ and $\bar{B}$. The second condition in Def. 6 (Def. 7) states that F_{i+1} (B_{i+1}) is an over-approximation of the post(pre)-image of F_i (B_i) w.r.t. M. We conclude that F_i over-approximates the set of states reachable from *INIT* in i steps, and B_i over-approximates the set of states reaching a violation of p in i steps. The following properties hold for FRS and BRS:

```
1: function DAR(M,p)
2:     if INIT ∧ ¬p == SAT then
3:         return cex
4:     end if
5:     F̄ = ⟨F_0 = INIT⟩, B̄ = ⟨B_0 = ¬p⟩
6:     n = 0
7:     while !F̄.FIXPOINT()∧!B̄.FIXPOINT() do
8:         if LOCSTRENGTHEN(F̄, B̄, n) == false then
9:             if GLBSTRENGTHEN(F̄, B̄, n) == false then
10:                return cex
11:            end if
12:        end if
13:        n = n + 1
14:    end while
15:    return Verified
16: end function
```

Fig. 1: Dual Approximated Reachability

Lemma 2. *A FRS (BRS) of length n exists iff there is no counterexample of length $\leq n$.*

Definition 8 (Fixpoint). *A FRS $\bar{F}_{[n]}$ is at* fixpoint *if there is $0 < k \leq n$ s.t. $F_k \Rightarrow \bigvee_{i=0}^{k-1} F_i$. Similarly, a BRS $\bar{B}_{[n]}$ is at* fixpoint *if there is $0 < k \leq n$ s.t. $B_k \Rightarrow \bigvee_{i=0}^{k-1} B_i$.*

Lemma 3. *Given a FRS $\bar{F}$ and a BRS $\bar{B}$, if $\bar{F}$ or $\bar{B}$ is at fixpoint then $M \models AGp$.*

Note that a fixpoint in one of the sequences suffices to conclude that $M \models AGp$.

4 Dual Approximated Reachability

In this section we describe our Dual Approximated Reachability (DAR) algorithm for model checking safety properties. DAR computes over-approximated sets of reachable states for both forward and backward reachability analysis by means of a FRS and a BRS, using interpolants. The computations are intertwined where each of them is used to make the other tighter. DAR avoids unrolling of the transition system unless it is really needed.

Technically, DAR computes a FRS $\bar{F}$ and a BRS $\bar{B}$ and gradually extends them until either a counterexample is found or a fixpoint is reached on either $\bar{F}$ or $\bar{B}$. Since the state-space of M is finite, one of the above is bound to happen, which ensures that:

Theorem 1. *Given a model M and a safety property $\varphi = AGp$, DAR always terminates. Moreover, $M \models \varphi$ if and only if DAR returns "Verified".*

We now describe DAR in detail. The pseudocode of DAR appears in Fig. 1.

Initialization of DAR (lines 2-5) starts by checking the formula $INIT \wedge \neg p$. If this formula is unsatisfiable, the initial states of M satisfy the property. If not, a counterexample exists. In the former case, DAR initializes $\bar{F} = \langle F_0 = INIT \rangle$ and $\bar{B} = \langle B_0 = \neg p \rangle$. Clearly $\bar{F}$ and $\bar{B}$ are FRS and BRS, respectively.

The iterative part of DAR (lines 8-13) then gradually extends and strengthens $\bar{F}$ and $\bar{B}$ s.t. they remain a FRS and a BRS respectively. As ensured by Lemma 2, this is possible as long as no counterexample of the corresponding length exists. In the following, we describe a single iteration of DAR, strengthening and extending $\bar{F}$ and $\bar{B}$, or reporting a counterexample.

4.1 First Iteration of DAR

Let us first present the first iteration of DAR. Recall that initially $\bar{F} = \langle F_0 = \mathit{INIT} \rangle$ and $\bar{B} = \langle B_0 = \neg p \rangle$. DAR then checks the formula $F_0 \wedge \mathit{TR} \wedge B_0' = \mathit{INIT} \wedge \mathit{TR} \wedge \neg p'$ for satisfiability. In case this formula is satisfiable a counterexample of length one exists. Otherwise, the unsatisfiability of $\mathit{INIT} \wedge \mathit{TR} \wedge \neg p'$ entails information both about the post-image of *INIT* and about the pre-image of $\neg p$. Accordingly, we extend $\bar{F}$ with $F_1 = \mathrm{FI}(F_0, B_0)$ and $\bar{B}$ with $B_1 = \mathrm{BI}(F_0, B_0)$. Due to the properties of the interpolants, the sequences $\bar{F} = \langle F_0, F_1 \rangle$ and $\bar{B} = \langle B_0, B_1 \rangle$ are FRS and BRS respectively.

4.2 General Iteration of DAR

Let us now discuss a general iteration $n+1$. Consider the FRS $\bar{F}_{[n]} = \langle F_0, F_1, \ldots, F_n \rangle$ and the BRS $\bar{B}_{[n]} = \langle B_0, B_1, \ldots B_n \rangle$ obtained at iteration n. The goal of iteration $n+1$ is to check if a counterexample of length $n+1$ exists, and if not, extend these sequences to length $n+1$ s.t. they remain a FRS and a BRS.

The combination of $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ provides an approximate description of all possible counterexamples of length $n+1$ in M. Namely, recall that F_i over-approximates the set of all states reachable from *INIT* in i steps. Similarly, B_j over-approximates the set of all states that can reach $\neg p$ in j steps. Their intersection, $F_i \wedge B_j$ therefore over-approximates the set of all states that are both reachable from *INIT* in i steps and can reach $\neg p$ in j steps. These are states that appear in the i-th step of a counterexample of length $i+j$. In particular, when we align $\bar{F}$ and $\bar{B}$ one against the other, conjoining F_i with B_{n-i+1}, we obtain an over-approximation of the set of all states that appear in the i-th step of a counterexample of length $n+1$. The sequence

$$\Pi(\bar{F}_{[n]}, \bar{B}_{[n]}) = \langle \mathit{INIT}, F_1 \wedge B_n, F_2 \wedge B_{n-1}, \ldots, F_n \wedge B_1, \neg p \rangle$$

therefore over-approximates the set of *all* counterexamples of length $n+1$.

We refer to the sequence $\Pi(\bar{F}_{[n]}, \bar{B}_{[n]})$ as an *approximated Counterexample* (aCEX). Whenever clear from the context we write Π and refer to the i-th element in the sequence as Π_i. A sequence of states $s_0, \ldots, s_{n+1}$ in M *matches* Π if for every $0 \leq i \leq n+1$, $s_i \in \Pi_i$. Formally, Π has the following property.

Lemma 4. *Let $\pi = s_0, \ldots, s_{n+1}$ be a counterexample in M. Then, π matches Π.*

By Lemma 4, checking if a counterexample exists amounts to checking if some path matches Π. Such a path is necessarily a counterexample of length $n+1$. If such a path exists, we say that Π is *valid*.

DAR first attempts to check for (in)validity of the aCEX using local checks in a *local strengthening phase*. If this fails, DAR moves on to the *global strengthening phase* that

applies global checks. In both phases, if the invalidity of the aCEX is established, the FRS and BRS are strengthened and extended into a FRS and a BRS of length $n+1$. Otherwise, the aCEX is found to be valid and a counterexample of length $n+1$ is obtained in the process.

Local Strengthening Phase The local strengthening phase aims at checking if Π is *locally invalid*, which provides a sufficient condition for its invalidity.

Definition 9. *Π is* locally invalid *if there exists $0 \leq i \leq n$ s.t. $\Gamma(\Pi_i, \Pi_{i+1}) \equiv \bot$.*

Lemma 5. *If Π is locally invalid, then it is also invalid.*

In order to check if Π is locally invalid, we use the following observation.

Lemma 6. *Let $\bar{F}_{[n]}$ be a FRS, $\bar{B}_{[n]}$ be a BRS, and $1 \leq i \leq n$. Then $\Gamma(F_i \wedge B_{n-i+1}, F_{i+1} \wedge B_{n-i}) \equiv \Gamma(F_i, B_{n-i})$. Similarly, $\Gamma(\text{INIT}, F_1 \wedge B_n) \equiv \Gamma(F_0, B_n)$, and $\Gamma(F_n \wedge B_1, \neg p) \equiv \Gamma(F_n, B_0)$. We conclude that for every $0 \leq i \leq n$, $\Gamma(\Pi_i, \Pi_{i+1}) \equiv \bot$ iff $\Gamma(F_i, B_{n-i}) \equiv \bot$.*

Lemma 6 follows from the property of a FRS, where $F_i \wedge TR \Rightarrow F'_{i+1}$, and the property of a BRS, where $B_{n-i+1} \Leftarrow TR \wedge B'_{n-i}$. Lemma 6 implies that if there exists $0 \leq i \leq n$ s.t. $\Gamma(F_i, B_{n-i}) \equiv \bot$, then the aCEX is locally invalid and hence invalid. This can also be understood intuitively, as the above means that the (over-approximated) set of states reachable from *INIT* in i steps and the (over-approximated) set of states that can reach $\neg p$ in $n-i$ steps are not reachable from one another in one step. This means that altogether $\neg p$ is not reachable from *INIT* in $i + (n-i) + 1 = n+1$ steps, and hence no counterexample of length $n+1$ exists.

In the local strengthening phase, DAR therefore searches for an index $0 \leq i \leq n$ s.t. $\Gamma(F_i, B_{n-i}) \equiv \bot$. It starts by checking the formula $\Gamma(F_n, B_0)$, setting $i = n$. In case it is satisfiable, DAR starts to iteratively go backwards along $\bar{F}$ and $\bar{B}$ decreasing i by 1. The traversal continues until either $\Gamma(F_i, B_{n-i})$ turns out to be unsatisfiable for some $0 \leq i \leq n$ or until $\Gamma(F_0, B_n)$ is found to be satisfiable.

If an index i is found s.t. $\Gamma(F_i, B_{n-i}) \equiv \bot$, then the aCEX is locally invalid and by Lemma 5 we conclude that no counterexample of length $n+1$ exists. Moreover, in this case, the FRS and BRS are locally and gradually strengthened and extended as follows.

Iterative Local Strengthening: Iterative local strengthening is reached when it is already known that no counterexample of length $n+1$ exists. Thus, as Lemma 2 ensures, there exist a FRS and BRS of length $n+1$. However, $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ cannot necessarily be extended immediately. For example, if $\Gamma(F_n, B_0) = F_n(V) \wedge TR(V, V') \wedge \neg p(V') \not\equiv \bot$, then no F_{n+1} can be obtained s.t. $F_n(V) \wedge TR(V, V') \Rightarrow F_{n+1}(V')$ and in addition $F_{n+1} \Rightarrow p$. On the other hand, if $\Gamma(F_n, B_0) \equiv \bot$ then F_{n+1} can be initialized using $\text{FI}(F_n, B_0)$ while maintaining the properties of a FRS (similarly to the initialization of F_1). Dually, if $\Gamma(F_0, B_n) \not\equiv \bot$, then no extension of $\bar{B}_{[n]}$ is possible, while if $\Gamma(F_0, B_n) \equiv \bot$, we can set $B_{n+1} = \text{BI}(F_0, B_n)$. We therefore first strengthen the components of $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ until $\Gamma(F_n, B_0) \equiv \bot$ and $\Gamma(F_0, B_n) \equiv \bot$, which is a necessary and sufficient condition for extending $\bar{F}$ and $\bar{B}$.

Recall that $\Gamma(F_i, B_{n-i}) \equiv \bot$ for some $0 \leq i \leq n$. This means that even though the components of $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ may not be precise enough to enable their extension, they are precise enough at least in one place that allowed us to conclude that no counterexample of length $n+1$ exists. DAR uses this "local" precision to strengthen the entire sequences, as described below.

In order to simplify the references to the indices, we replace the use of i and $n-i$ by $0 \leq i, j \leq n$ s.t. $i + j = n$. Therefore $\Gamma(F_i, B_j) \equiv \bot$ for some $0 \leq i, j \leq n$ s.t. $i + j = n$. This ensures that there exists a forward interpolant $\mathrm{FI}(F_i, B_j)$, as well as a backward interpolant $\mathrm{BI}(F_i, B_j)$. We can therefore perform a *local strengthening step* updating F_{i+1} and B_{j+1}:

Definition 10. *Let $\bar{F}_{[n]}$ be a FRS and $\bar{B}_{[n]}$ be a BRS s.t. $\Gamma(F_i, B_j) \equiv \bot$ for some $0 \leq i, j \leq n$ s.t. $i + j = n$. A* forward strengthening step at (i, j) *strengthens $\bar{F}_{[n]}$: If $i < n$, $F_{i+1} = F_{i+1} \wedge \mathit{FI}(F_i, B_j)$. A* backward strengthening step at (i, j) *strengthens $\bar{B}_{[n]}$: If $j < n$, $B_{j+1} = B_{j+1} \wedge \mathit{BI}(F_i, B_j)$.*

We refer to $i, j < n$ since F_{n+1} and B_{n+1} are not yet defined and therefore cannot be updated. The strengthening propagates the unsatisfiability of $\Gamma(F_i, B_j)$ one step forward and one step backward while maintaining the properties of a FRS and a BRS:

Lemma 7. *Let $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ be the result of a forward or backward strengthening step at (i, j) s.t. $i + j = n$. Then $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$ remain FRS and BRS resp. In addition:*

- *For a forward strengthening step, if $i < n$, $\Gamma(F_{i+1}, B_{j-1}) \equiv \bot$.*
- *For a backward strengthening step, if $j < n$, $\Gamma(F_{i-1}, B_{j+1}) \equiv \bot$.*

Lemma 7 implies that if $\Gamma(F_i, B_j) \equiv \bot$ for *some* $0 \leq i, j \leq n$ s.t. $i+j = n$, then by iterating the forward and backward strengthening steps, we can eventually ensure that $\Gamma(F_i, B_j) \equiv \bot$ for *every* $0 \leq i, j \leq n$ s.t. $i + j = n$, and in particular for $i = 0, j = n$ and $i = n, j = 0$. Thus, we apply an *iterative local strengthening* starting from (i, j):

Definition 11 (Iterative Local Strengthening). *Let $0 \leq i, j \leq n$ be indices s.t. $i+j = n$ and $\Gamma(F_i, B_j) \equiv \bot$.* Iterative local strengthening from (i, j) *performs:*

1. *Forward strengthening steps starting at (i, j), proceeding forward while increasing i and decreasing j until $(n-1, 1)$ (strengthening $F_{i+1}, \ldots, F_n$), and*
2. *Backward strengthening steps starting at (i, j), proceeding backward while increasing j and decreasing i until $(1, n-1)$ (strengthening $B_{j+1}, \ldots, B_n$), and*
3. *Finally, once $\Gamma(F_n, B_0) \equiv \bot$, F_{n+1} is initialized by $\mathit{FI}(F_n, B_0)$. Similarly, once $\Gamma(F_0, B_n) \equiv \bot$, B_{n+1} is initialized by $\mathit{BI}(F_0, B_n)$.*

Lemma 8. *Let $0 \leq i, j \leq n$ be indices s.t. $i + j = n$ and $\Gamma(F_i, B_j) \equiv \bot$. Iterative local strengthening from (i, j) terminates with a FRS and a BRS of length $n + 1$.*

Iterative local strengthening uses the BRS for the strengthening of the FRS and vice versa, demonstrating how each of them is used to make the other over-approximation tighter. The complete local strengthening procedure is described in Fig. 2.

```
17: function LocStrengthen(F̄, B̄, n)
18:     i = FindStrengthen(F̄, B̄, n)
19:     if i == −1 then
20:         // No local strengthening
21:         // point was found
22:         //Move to GlbStrengthen
23:         return false
24:     else
25:         IterLS(F̄, B̄, n, i, n − i)
26:         return true
27:     end if
28: end function
```

(a) Local Strengthening

```
29: function IterLS(F̄, B̄, n, i, j)
30:     while i < n do
31:         F_{i+1} = F_{i+1} ∧ FI(F_i, B_{n−i})
32:         i = i + 1
33:     end while
34:     F̄.Add(FI(F_n, B_0))
35:     while j < n do
36:         B_{j+1} = B_{j+1} ∧ BI(F_{n−j}, B_j)
37:         j = j + 1
38:     end while
39:     B̄.Add(BI(F_0, B_n))
40: end function
```

(b) Iterative Local Strengthening

Fig. 2: Local strengthening procedures

Global Strengthening Phase We now consider the case where $\Gamma(F_i, B_{n-i}) \not\equiv \bot$ for every $0 \leq i \leq n$ in $\bar{F}_{[n]}$ and $\bar{B}_{[n]}$. By Lemma 6, this means that there is a real transition between every pair of consecutive sets in the aCEX Π, making local strengthening inapplicable since the aCEX is not locally invalid. Clearly this does not imply that the aCEX is valid, and further checks are needed. We therefore turn to examine the (in)validity of the aCEX in a more global manner.

Similarly to the principle used in CEGAR [5] for an *abstract* counterexample, here too, if the aCEX Π is invalid, there exists a minimal index $i \leq n+1$ representing the minimal prefix of the aCEX that has no matching path in M. We therefore wish to search for such an index, if it exists. The search starts from the prefix Π_0, Π_1, Π_2 (since $\langle \Pi_0, \Pi_1 \rangle$ is necessarily valid) and extends it gradually. In the i-th step (starting from $i = 2$), the goal is to check if $\Pi_0 \wedge \mathit{TR} \wedge \Pi_1' \wedge \mathit{TR} \wedge \Pi_2'' \ldots \wedge \mathit{TR} \wedge \Pi_i^{\langle i \rangle}$ (*) is satisfiable, meaning that a matching path to the prefix $\Pi_0, \ldots, \Pi_i$ exists in M.

Recall that for $i \leq n$, (*) is actually the formula $\mathit{INIT} \wedge \mathit{TR} \wedge (F_1 \wedge B_n)' \wedge \mathit{TR} \wedge (F_2 \wedge B_{n-1})'' \wedge \ldots \wedge \mathit{TR} \wedge (F_i \wedge B_{n-i+1})^{\langle i \rangle}$. For $i = n+1$ the last conjunct consists of B_0 only (without an $\bar{F}$-component). In fact, since in a FRS $F_j \wedge \mathit{TR} \Rightarrow F_{j+1}'$, then removing all $\bar{F}$ components except for the first (*INIT*) results in an equivalent formula. Similarly, since in a BRS $B_{j+1} \Leftarrow \mathit{TR} \wedge B_j'$, removing all $\bar{B}$ components but the last (B_{n-i+1}) again results in an equivalent formula. This simplifies the formula as follows.

Lemma 9. *For every* $2 \leq i \leq n+1$*:* $\Pi_0 \wedge \mathrm{TR} \wedge \Pi_1' \wedge \mathrm{TR} \wedge \Pi_2'' \wedge \ldots \wedge \mathrm{TR} \wedge \Pi_i^{\langle i \rangle}$ *is equivalent to* $\mathrm{INIT} \wedge \mathrm{TR} \wedge \mathrm{TR} \wedge \ldots \wedge \mathrm{TR} \wedge B_{n-i+1}^{\langle i \rangle}$.

DAR therefore checks formulas of the form $\mathit{INIT} \wedge \mathit{TR} \wedge \ldots \wedge \mathit{TR} \wedge B_{n-i+1}^{\langle i \rangle}$ starting from $i = 2$. It keeps on adding transitions until either the formula becomes unsatisfiable, or until $i = n+1$ is reached (ending with $B_0 = \neg p$). If the formula is still satisfiable for $i = n+1$, a counterexample is found and DAR terminates.

If for some $2 \leq i \leq n+1$, $\mathit{INIT} \wedge \mathit{TR} \wedge \ldots \wedge \mathit{TR} \wedge B_{n-i+1}^{\langle i \rangle}$ turns out to be unsatisfiable, making the aCEX invalid, then first $\bar{F}_{[n]}$ is strengthened:

```
41: function GLBSTRENGTHEN(F̄, B̄, n)
42:     for i = 2 → n + 1 do        // n = 0 does not go into the loop
43:         if INIT ∧ TR ... ∧ TR ∧ B̄^⟨i⟩_{n−i+1} == UNSAT then
44:             Ī = GETINTERPOLATIONSEQ()
45:             for j = 1 → min{i, n} do
46:                 F_j = F_j ∧ I_j
47:             end for
48:             ITERLS(F̄, B̄, n, i − 1, n − i + 1)
49:             return true
50:         end if
51:     end for
52:     return false        // counterexample
53: end function
```

Fig. 3: Global strengthening procedure

Definition 12. *Let* INIT $\wedge$ TR $\wedge \ldots \wedge$ TR $\wedge B^{\langle i \rangle}_{n-i+1} \equiv \bot$ *for some* $2 \leq i \leq n+1$, *and let* $\langle I_0, I_1, \ldots, I_{i+1} \rangle$ *be an interpolation-sequence for* $\langle A_1 = \text{INIT} \wedge \text{TR}, A_2 = \text{TR}, \ldots, A_i = \text{TR}, A_{i+1} = B^{\langle i \rangle}_{n-i+1} \rangle$. *A* global strengthening step at index i *strengthens* F_j *for every* $1 \leq j \leq \min\{i, n\}$ *by setting* $F_j = F_j \wedge I_j$.

The condition $1 \leq j \leq \min\{i, n\}$ ensures that if $i = n+1$, strengthening is applied only up to F_n since F_{n+1} is not yet defined[3]. The following Lemma, along with Lemma 6 ensures that after a global strengthening step, the strengthened aCEX is locally invalid.

Lemma 10. *Let* $\bar{F}_{[n]}$ *be the result of a global strengthening step at index* $2 \leq i \leq n+1$. *Then* $\bar{F}_{[n]}$ *remains a FRS. In addition,* $\Gamma(F_{i-1}, B_{n-i+1}) \equiv \bot$.

DAR now uses iterative local strengthening from $(i-1, n-i+1)$ (Def. 11) to strengthen $F_i, \ldots, F_n$ and $B_{n-i+2}, \ldots, B_n$[4], as well as initialize F_{n+1} and B_{n+1}. The complete global strengthening procedure is described in Fig. 3.

5 Experimental Results

To implement DAR we collaborated with *Jasper Design Automation*[5]. We measured the efficiency of DAR by comparing it against two top-tier methods: ITP and IC3. We used Jasper's formal verification platform in order to implement DAR, ITP and IC3. Collaborating with Jasper allowed us to experiment with various real-life industrial designs and properties from various major semiconductor companies.

[3] If a global strengthening step is performed at $i = n+1$, then F_{n+1} can be initialized to I_{n+1}.

[4] Note that instead of performing a local strengthening of $\bar{B}$ as part of the iterative local strengthening, an interpolation-sequence $\langle J_0, J_1, \ldots, J_{i+1} \rangle$ for $\langle A_1 = \mathit{TR} \wedge B^{\langle i+1 \rangle}_{n-i}, A_2 = \mathit{TR}, \ldots A_i = \mathit{TR}, A_{i+1} = \mathit{INIT} \rangle$ can be used to strengthen $B_{n-i+1}, \ldots, B_n$ by setting $B_{n-i+j} = B_{n-i+j} \wedge J_j$ for $1 \leq j \leq i$, and to initialize B_{n+1} to J_{i+1}. In this case, iterative local strengthening will be performed only forward, updating $\bar{F}$ only. For simplicity of the presentation, we use iterative local strengthening both forward and backward instead of using an interpolation-sequence for the backward update.

[5] An EDA company: http://www.jasper-da.com

Table 1: Parameters of the experiments. *Name*: name of the property; ♯*Vars*: number of state variables in the cone of influence; *Status*: *true* - verified property, *false* - indicates a counterexample; *D*: *convergence depth* representing the number of over-approximated sets of states computed when the algorithm converges (for ITP, the number of sets computed for the last bound used, and for DAR, the length of $\bar{F}$ and $\bar{B}$); *MaxU*: *maximum unrolling* used during verification; ♯*GS*: number of times Global Strengthening is used in DAR; GS_R: ratio between iterations using global strengthening to the total number of iterations; *Time[s]*: *time* in seconds. Minimal runtime appears in boldface. Properties above the full line are from real industrial designs. The rest are from HWMCC'11.

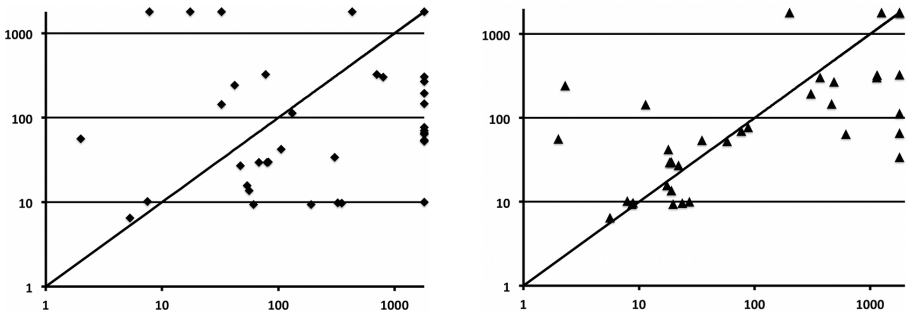
			IC3		ITP			DAR				
Name	♯Vars	Status	D	Time[s]	D	MaxU	Time[s]	D	MaxU	♯GS	GS_R	Time[s]
Ind_1	11854	true	46	799	41	28	1138	49	35	21	0.42	**303**
Ind_2	11854	true	44	701	41	28	1148	49	35	18	*0.36*	**326**
Ind_3	11866	true	11	82	5	2	**19.1**	11	8	4	*0.33*	29.9
Ind_4	11877	true	NA	TO	33	12	307	36	30	18	0.48	**194**
Ind_5	11871	false	NA	TO	NA	20	88	19	20	10	0.5	**77**
Ind_6	11843	false	NA	TO	NA	19	77	18	19	9	0.47	**70**
Ind_7	1247	true	6	**1.5**	3	2	2	17	5	9	0.5	56.3
Ind_8	1247	true	7	**7.8**	17	23	1250	NA	NA	NA	NA	TO
Ind_9	449	true	337	**78**	NA	NA	TO	45	12	22	0.48	327
Ind_{10}	331	true	458	305	NA	NA	TO	26	11	15	0.56	**33.9**
Ind_{11}	330	true	419	132	NA	NA	TO	38	12	19	0.49	**113**
Ind_{12}	450	true	22	**32.5**	NA	NA	TO	NA	NA	NA	NA	TO
Ind_{13}	3837	false	NA	TO	NA	68	369	67	68	33	0.48	**305**
Ind_{14}	3837	false	NA	TO	NA	69	487	68	69	25	*0.36*	**269**
Ind_{15}	3836	true	6	42	4	2	**2.3**	70	64	32	0.45	243
Ind_{16}	11860	true	9	32.5	5	2	**11.4**	33	32	16	0.47	144
Ind_{17}	11878	true	14	68	7	4	**18.4**	11	8	4	*0.33*	29.5
Ind_{18}	3836	true	NA	TO	6	17	27.3	15	6	6	*0.37*	**10**
intel007	1307	true	5	**53.5**	NA	NA	TO	NA	NA	NA	NA	TO
intel018	491	true	NA	TO	57	35	695	78	51	33	0.42	**64**
intel019	510	true	NA	TO	52	35	515	96	57	43	0.44	**310**
intel023	358	true	NA	TO	NA	NA	TO	86	53	35	*0.4*	**66**
intel026	492	true	53	47.1	50	35	**21.9**	70	51	34	0.48	27.8

Our implementations use known optimizations for the checked methods (e.g. [2,9]) and are comparable to other optimized implementations available online. For DAR we used some basic procedures to simplify the computed interpolants when possible. Our implementation of DAR is preliminary and can be further optimized.

For the experiments we used 37 real safety properties from real industrial hardware designs. The timeout was set to 1800 seconds and experiments were conducted on systems with Intel Xeon X5660 running at 2.8GHz and 24GB of main memory.

Table 1 shows different parameters for all three algorithms on various industrial examples. *Time* and *convergence depth* are presented for all three, whereas *maximum unrolling* is presented only for ITP and DAR (IC3 does not use unrolling). For DAR we also present ♯GS and GS_R that refer to *global strengthening* (using unrolling) and indicate the number, and ratio, of iterations where *local strengthening* was insufficient.

Examining the results shows that the use of unrolling in DAR is indeed limited and that *local strengthening* plays a major part during verification, with $\text{GS}_R < 0.5$ in most cases, indicating that local strengthening is often sufficient. Moreover, even when unrolling is used, its depth is usually smaller compared to the convergence depth, as



(a) Runtime DAR vs. IC3. (b) Runtime DAR vs. ITP.

Fig. 4: Y-axis represents DAR's runtime in seconds. X-axis represents runtime in seconds for the compared algorithm (IC3 or ITP). Points below the diagonal are in favor of DAR.

indicated by maximum unrolling. Note that the maximum unrolling provides an *upper bound* on the unrolling, and the actual unrolling can be smaller in some global strengthening phases. For falsified properties (counterexample exists) unrolling is necessarily applied up to the length of the counterexample in the last iteration. Yet, in many cases local strengthening is still sufficient in previous iterations.

Another conclusion from the table is that a lower depth of convergence does not necessarily translate to a better runtime. We can see that in many cases, while ITP converges with less computed sets it takes more time than DAR. This is not surprising since the number of computed sets presented for ITP considers only the sets computed in the last bound that was used, disregarding sets from previous bounds. The same can be seen with regards to IC3. While IC3 converges at a lower depth (on some cases), it still does not necessarily perform better. This is mainly due to the different effort invested by each algorithm in the strengthening and addition of a new over-approximated set.

Fig. 4 shows a runtime comparison between DAR and IC3 (Fig. 4a) and ITP (Fig. 4b) on all 37 industrial examples, including those from Table 1. In 19 out of 37 cases, DAR outperforms ITP, and in 25 out of 37 cases it outperforms IC3. In 18 out of 37 cases DAR outperforms both methods. DAR could not solve only 5 cases, whereas ITP and IC3 failed to solve 7 and 12 cases respectively. The overall performance, when summarized, is in favor of DAR with 36% improvement in run time when compared to ITP and 52% improvement when compared to IC3.

Cases where DAR outperforms ITP can be explained by the following factors. First, DAR avoids unrolling when not needed, therefore its SAT calls are simpler. Second, DAR uses over-approximated sets computed in early iterations and strengthens them as needed, while ITP does not re-use sets that were computed for lower bounds and restarts its computation when a spurious counterexample is encountered. Cases where DAR outperforms IC3 are typically when DAR's strengthening is more efficient than IC3's inductive generalization, requiring less computation power at each iteration.

Since DAR relies heavily on interpolants, the cases where DAR performs worse than IC3 are usually those where the interpolants grow large and contain redundancies. This is also true when comparing to ITP. Since DAR computes more interpolants than ITP and also accumulates them, it is more sensitive to the size of the computed interpolants.

We also used the HWMCC'11 benchmark in our experiments. While there are a lot of cases where all methods perform the same, there are also examples where DAR outperforms both IC3 and ITP (some are shown at the bottom of Table 1). The benchmark also includes examples where IC3 or ITP perform better than DAR. The majority of these cases are simple and solved in a few seconds.

6 Conclusions

We present DAR, a complete SAT-based model checking algorithm that uses both *forward* and *backward* interpolants to traverses the state space in a mostly local manner.

The experimental results show that DAR performs well on many industrial designs, and in many cases outperforms the successful ITP and IC3 algorithms. These results are very encouraging, especially since our implementation of DAR can be optimized much further. For example, the local checks applied in the local strengthening phase are independent of each other, which makes DAR most suitable for a parallel implementation.

Our experiments were conducted on hardware designs. However, DAR is not restricted to hardware. It will be interesting to see how it performs on software systems.

Another possible direction for future work refers to an integration of DAR with lazy abstraction [15]. The fact that DAR maintains over-approximations of sets of states reachable from *INIT* or $\neg p$ in *exactly* i steps, rather than in *at most* i steps, enables more flexibility in the choice of abstraction used at each time frame.

References

1. A. R. Bradley. SAT-based model checking without unrolling. In *VMCAI*, 2011.
2. G. Cabodi, M. Murciano, S. Nocco, and S. Quer. Stepping forward with interpolants in unbounded model checking. In *ICCAD*, pages 772–778, 2006.
3. G. Cabodi, S. Nocco, and S. Quer. Mixing forward and backward traversals in guided-prioritized bdd-based verification. In *CAV*, pages 471–484, 2002.
4. G. Cabodi, S. Nocco, and S. Quer. Interpolation sequences revisited. In *DATE*, pages 316–322, 2011.
5. E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement for symbolic model checking. *JACM'03*.
6. E. C. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT press, 1999.
7. W. Craig. Linear reasoning. a new form of the herbrand-gentzen theorem. *J. Symb. Log.*, 22(3), 1957.
8. V. D'Silva, M. Purandare, and D. Kroening. Approximation refinement for interpolation-based model checking. In *VMCAI*, pages 68–82, 2008.
9. N. Een, A. Mishchenko, and R. Brayton. Efficient implementation of property directed reachability. In *FMCAD*, 2011.
10. R. Jhala and K. McMillan. Interpolant-based transition relation approximation. In *CAV'05*.
11. K. L. McMillan. Interpolation and SAT-based Model Checking. In *CAV*, 2003.
12. K. L. McMillan. Lazy Abstraction with Interpolants. In *CAV*, 2006.
13. C. Stangier and T. Sidle. Invariant checking combining forward and backward traversal. In *FMCAD*, pages 414–429, 2004.
14. Y. Vizel and O. Grumberg. Interpolation-sequence based model checking. In *FMCAD*, 2009.
15. Y. Vizel, O. Grumberg, and S. Shoham. Lazy abstraction and SAT-based reachability in hardware model checking. In *FMCAD*, 2012.