Control-flow Guided Clause Generation for Property Directed Reachability

Xian Li, Klaus Schneider

Embedded Systems Group
Department of Computer Science
University of Kaiserslautern, Germany

HLDVT 2016 at Santa Cruz, California, U.S.A., October 7 – 8, 2016
### Table of Contents

1. Motivation

2. Property Directed Reachability

3. Control-flow Guided Clause Generation for PDR

4. Summary
## Outline

1. Motivation
2. Property Directed Reachability
3. Control-flow Guided Clause Generation for PDR
4. Summary
Formal Verification of Synchronous Hardware Circuits

- PDR: a very efficient verification method based on induction

Synchronous Circuits

A four-bit "up" counter

\[ \varphi \]
Formal Verification of Synchronous Programs

- PDR: a very efficient verification method based on induction

**Synchronous Programs**

```
module M(event bool ?a, ?b, o1, o2) {
  loop {
    l1: pause;
    if(o1 & (a|b)) {
      emit(o2);
      l2: await(a);
    }
  }
}
```

**Synchronous Circuits**

A four-bit "up" counter

\[ \varphi \]

\[ \varphi' \]
Exploit control-flow of synchronous languages to improve the performance of the PDR method.

- check the unreachability of counterexamples to induction (CTIs)
- generalize CTIs to exclude as many unreachable states as possible
Outline

1. Motivation

2. Property Directed Reachability

3. Control-flow Guided Clause Generation for PDR

4. Summary
Property Directed Reachability

Target: Prove \( \Phi \) is valid w.r.t. \( \mathcal{K} \)
- a state transition system: \( \mathcal{K} := (\mathcal{V}, \mathcal{I}, \mathcal{T}) \)
- a safety property: \( \Phi \)
- \( \Phi \) holds on all reachable states of \( \mathcal{K} \)

\[
\begin{align*}
\mathcal{V} & := \{a, b, c\} \\
\mathcal{I} & := \neg(a \lor b \lor c) \\
\mathcal{T} & := \neg(a \lor b \lor c) \land (a' \land \neg b' \land \neg c') \\
& \lor (a \land \neg b \land \neg c) \land (\neg a' \land \neg b' \land c') \\
& \lor \ldots \\
\Phi & := \neg a \lor b \lor \neg c
\end{align*}
\]
Property Directed Reachability

Target: Prove $\Phi$ is valid w.r.t. $\mathcal{K}$
  - a state transition system: $\mathcal{K} := (V, I, T)$
  - a safety property: $\Phi$
  - $\Phi$ holds on all reachable states of $\mathcal{K}$

$\Phi$ is inductive w.r.t. $\mathcal{K}$
  - induction base: $\Phi$ holds in all initial states
  - induction step: $\lceil \Phi \rceil_{\mathcal{K}}$ have no successor violating $\Phi$
Property Directed Reachability

Target: Prove $\Phi$ is valid w.r.t. $\mathcal{K}$

- a state transition system: $\mathcal{K} := (\mathcal{V}, \mathcal{I}, \mathcal{T})$
- a safety property: $\Phi$
- $\Phi$ holds on all reachable states of $\mathcal{K}$

$\Phi$ is inductive w.r.t. $\mathcal{K}$

- induction base: $\Phi$ holds in all initial states
- induction step: $[\Phi]_{\mathcal{K}}$ have no successor violating $\Phi$
Property Directed Reachability

PDR method constructs a sequence of clause sets $\Psi_0, \ldots, \Psi_k$ that overapproximate the states are reachable in $0, \ldots, k$ steps.

- **propagation**: extend the sequence $\Psi_0, \ldots, \Psi_k$
- **blocking**: narrow the overapproximation $\Psi_k$
PDR method constructs a sequence of clause sets $\Psi_0, \ldots, \Psi_k$ that overapproximate the states are reachable in $0, \ldots, k$ steps.

- **propagation**: extend the sequence $\Psi_0, \ldots, \Psi_k$
- **blocking**: narrow the overapproximation $\Psi_k$
Blocking Phase

\[ \psi_{k-1} = \text{Clause}(I) \]

\[ \psi_k = \psi_\Phi = \text{Clause}(\Phi) \]
Motivation

Property Directed Reachability

Control-flow Guided Clause Generation for PDR

Summary

Blocking Phase

- $\psi_{k-1} = \text{Clause}(I)$
- $\psi_k = \psi_\phi = \text{Clause}(\phi)$

Whether $\psi_k$ has successors violating $\phi$? Yes

SAT solver returns a CTI, in the form of a cube $C_k$, i.e., a conjunction of literals over $V$.
Blocking Phase

- CTI generalization
- reachability of CTI
- clause generalization to narrow $\Psi_k$
Blocking Phase

- CTI generalization
- reachability of CTI
- clause generalization to narrow $\Psi_k$

![Diagram showing states $s_0$ to $s_7$ with states colored to indicate whether $\phi$ holds or doesn't hold, and Reachable States highlighted in yellow.](image-url)
Blocking Phase

- CTI generalization
- reachability of CTI
- clause generalization to narrow $\Psi_k$
Blocking Phase

- CTI generalization
- reachability of CTI
- clause generalization to narrow $\Psi_k$
1. Motivation

2. Property Directed Reachability

3. Control-flow Guided Clause Generation for PDR

4. Summary
Exploit control-flow of synchronous languages to improve the performance of the blocking phase of PDR method.

- CTI generalization
- reachability of CTI
- clause generalization to narrow $\Psi_k$
Transition Systems of a Synchronous Program

Let $\mathcal{V} := \mathcal{V}^{cf} \cup \mathcal{V}^{df}$ and $\mathcal{K} := \mathcal{K}^{cf} \times \mathcal{K}^{df}$, with

- $\mathcal{K} = (\mathcal{V}, \mathcal{I}, \mathcal{T})$
- $\mathcal{K}^{cf} = (\mathcal{V}, \mathcal{I}^{cf}, \mathcal{T}^{cf})$
- $\mathcal{K}^{df} = (\mathcal{V}, \mathcal{I}^{df}, \mathcal{T}^{df})$
Transition Systems of a Synchronous Program

Let $\mathcal{V} := \mathcal{V}^{\text{cf}} \cup \mathcal{V}^{\text{df}}$ and $\mathcal{K} := \mathcal{K}^{\text{cf}} \times \mathcal{K}^{\text{df}}$, with

- $\mathcal{K} = (\mathcal{V}, I, T)$
- $\mathcal{K}^{\text{cf}} = (\mathcal{V}, I^{\text{cf}}, T^{\text{cf}})$
- $\mathcal{K}^{\text{df}} = (\mathcal{V}, I^{\text{df}}, T^{\text{df}})$

unreachability of CTIs in $\mathcal{K}$ can be proved by unreachability in $\mathcal{K}^{\text{cf}}$
First Improvement for the Blocking Phase

Let $\mathcal{V} := \mathcal{V}^{\text{cf}} \cup \mathcal{V}^{\text{df}}$ and $\mathcal{K} := \mathcal{K}^{\text{cf}} \times \mathcal{K}^{\text{df}}$, with

- $\mathcal{K} = (\mathcal{V}, \mathcal{I}, \mathcal{T})$
- $\mathcal{K}^{\text{cf}} = (\mathcal{V}, \mathcal{I}^{\text{cf}}, \mathcal{T}^{\text{cf}})$
- $\mathcal{K}^{\text{df}} = (\mathcal{V}, \mathcal{I}^{\text{df}}, \mathcal{T}^{\text{df}})$

unreachability of CTIs in $\mathcal{K}$ can be proved by unreachability in $\mathcal{K}^{\text{cf}}$

- reachability of CTIs in $\mathcal{K}$
  - simpler unreachability tests in $\mathcal{K}^{\text{cf}}$
Let $\mathcal{V} := \mathcal{V}^{\text{cf}} \cup \mathcal{V}^{\text{df}}$ and $\mathcal{K} := \mathcal{K}^{\text{cf}} \times \mathcal{K}^{\text{df}}$, with

- $\mathcal{K} = (\mathcal{V}, \mathcal{I}, \mathcal{T})$
- $\mathcal{K}^{\text{cf}} = (\mathcal{V}, \mathcal{I}^{\text{cf}}, \mathcal{T}^{\text{cf}})$
- $\mathcal{K}^{\text{df}} = (\mathcal{V}, \mathcal{I}^{\text{df}}, \mathcal{T}^{\text{df}})$

unreachability in $\mathcal{K}^{\text{cf}}$ is independent on the dataflows
Let $\mathcal{V} := \mathcal{V}^{\text{cf}} \cup \mathcal{V}^{\text{df}}$ and $\mathcal{K} := \mathcal{K}^{\text{cf}} \times \mathcal{K}^{\text{df}}$, with

- $\mathcal{K} = (\mathcal{V}, \mathcal{I}, \mathcal{T})$
- $\mathcal{K}^{\text{cf}} = (\mathcal{V}, \mathcal{I}^{\text{cf}}, \mathcal{T}^{\text{cf}})$
- $\mathcal{K}^{\text{df}} = (\mathcal{V}, \mathcal{I}^{\text{df}}, \mathcal{T}^{\text{df}})$

unreachability in $\mathcal{K}^{\text{cf}}$ is independent on the dataflows

- generalize CTIs to narrow the reachable state approximations
  if $\mathcal{C}$ is unreachable, then generalize $\neg \mathcal{C}'$ instead of $\neg \mathcal{C}$:
  $\mathcal{C}' := \mathcal{C}' \mid_{\mathcal{V}^{\text{cf}}}$ obtained by omitting the dataflow literals in $\mathcal{C}$
Outline

1. Motivation
2. Property Directed Reachability
3. Control-flow Guided Clause Generation for PDR
4. Summary
Exploit control-flow of synchronous languages to improve the performance of the blocking phase of PDR method.

- reachability of CTIs in \( \mathcal{K} \)
  - simpler unreachability tests in \( \mathcal{K}^{cf} \)
- generalize CTIs to narrow the reachable state approximations
  - if \( C \) is unreachable, then generalize \( \neg C' \) instead of \( \neg C \): \( C' := C|_{\mathcal{V}^{cf}} \) obtained from omitting the dataflow literals in \( C \)
Thank you for your attention.

Questions?