IC3 Software Model Checking on Control Flow Automata

Tim Lange ¹  Martin R. Neuhäußer ²  Thomas Noll ¹

¹ Software Modeling and Verification Group, RWTH Aachen
² Siemens AG

FMCAD 2015 at Austin, TX, USA, September 29, 2015
Introduction

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
Introduction

Motivation

Lifting to software model checking

- IC3 had a deep impact in hardware model checking
- Showed much better performance than CEGAR and BMC
- Nowadays employed in most major hardware model checking tools

Challenges

- Domain in hardware model checking finite (bit-level)
- How to handle infinite state spaces?
- How to encode finite control flow?
Preliminaries

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
Preliminaries

Control Flow Automaton (CFA)

A CFA $\mathcal{A} = (L, G, l_0, l_E)$ consists of a set of locations $L = \{0, \ldots, n\}$ and edges in $G \subseteq L \times QFFO \times L$ labeled with quantifier-free first-order formulas, an initial location $l_0$, and an error location $l_E$.

Transition formula

Given two locations $l_1, l_2 \in L$, we define the transition formula

$$T_{l_1 \rightarrow l_2} = \begin{cases} (pc = l_1) \land t \land (pc' = l_2) & \text{, if } (l_1, t, l_2) \in G \\ false & \text{, otherwise.} \end{cases}$$
Preliminaries

Relative Inductivity

Given a transition formula $T = \bigvee_{(l_1,t,l_2) \in G} T_{l_1 \rightarrow l_2}$, a formula $\varphi$ is **inductive relative** to another formula $\psi$ if

$$\psi \land \varphi \land T \Rightarrow \varphi'$$

is valid.

Edge-Relative Inductivity

Given a CFA $A$ and locations $l_1, l_2 \in L$, a formula $\varphi$ is **inductive edge-relative** to another formula $\psi$ if

$$\psi \land \varphi \land T_{l_1 \rightarrow l_2} \Rightarrow \varphi'$$

is valid.

---

## Preliminaries

### Region

A region \( r = (l, s) \) is a pair consisting of location \( l \) and formula \( s \). The set of corresponding formulas for \( r \) is given as \( \{ \varphi \mid \varphi \equiv (pc = l \land s) \} \). Similarly, for \( \neg r \) corresponding formulas are defined as \( \{ \varphi \mid \varphi \equiv \neg(pc = l \land s) \} \).

### Edge-Relative Inductive Regions

Assume two regions \( r_1 = (l_1, s_1) \), \( \neg r_2 = \neg(l_2, s_2) \), we can reduce edge-relative inductivity of \( \neg r_2 \) to \( r_1 \) to

\[
\begin{align*}
  s_1 \land T_{l_1 \to l_2} \Rightarrow \neg s_2' &, \text{ if } l_1 \neq l_2 \\
  s_1 \land \neg s_2 \land T_{l_1 \to l_2} \Rightarrow \neg s_2' &, \text{ if } l_1 = l_2
\end{align*}
\]

---

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
Consider the transition system $\mathcal{M} = (X, I, T)$
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$.
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 

![Diagram of transition system and property]
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$.
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$.
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Consider the transition system $\mathcal{M} = (X, I, T)$ and the property $P(X)$. 
Related Work

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
## Related Work

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract reachability tree (ART) unrolling</td>
<td>[CG12]</td>
</tr>
<tr>
<td>Unroll ART, search error path and refute (similarly to blocking phase of IC3).</td>
<td></td>
</tr>
<tr>
<td>Bit-blasting</td>
<td>[WK13]</td>
</tr>
<tr>
<td>Encode variables as bit-vectors and use bit-blasting with bit-level IC3.</td>
<td></td>
</tr>
<tr>
<td>Implicit Abstraction</td>
<td>[Cim+14]</td>
</tr>
<tr>
<td>Express abstract transitions without explicitly computing the abstract system.</td>
<td></td>
</tr>
<tr>
<td>Predicate Abstraction</td>
<td>[BBW14]</td>
</tr>
<tr>
<td>Use predicate abstraction and refine predicates based on CTIs.</td>
<td></td>
</tr>
</tbody>
</table>

---


IC3 on Control Flow Automata

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
IC3 on Control Flow Automata

Idea

- Encoding of control flow using special \( pc \) variable not efficient \([CG12]\)
- Extraction of control flow advantageous
- Instead of unrolling into ART apply IC3 directly on CFA
- For every location in the CFA construct frames \( F_0, \ldots, F_k \)
- Frames represent overapproximations of \( i \)-step reachability in location
- Explicit control flow locations allow to take only single transitions into account

IC3 on Control Flow Automata

Example

Initial location: $l_0$
Error location: $l_E$
Terminating location: 2
Example

IC3 on Control Flow Automata

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i:$</th>
<th>$l_0$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>
IC3 on Control Flow Automata

Example

![Diagram]

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i:$</th>
<th>$l_0$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

CTI ($1, x \neq y$), level 1
IC3 on Control Flow Automata

Example

Frames $F_{i,l}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$l$</th>
<th>$l_0$</th>
<th>$1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>

CTI $(1, x \neq y)$, level 1

$SAT(F_{(0,1)} \land \neg(x \neq y) \land T_{1\rightarrow1} \land x' \neq y')$
IC3 on Control Flow Automata

Example

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$l$</th>
<th>$l_0$</th>
<th>$1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>

CTI $(1, x \neq y)$, level 1

$SAT(F_{(0,1)} \land \neg(x \neq y) \land T_{1 \rightarrow 1} \land x' \neq y') \times$
**Example**

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i:$</th>
<th>$l_0$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

CTI $(1,x \neq y)$, level 1

$SAT(F_{(0,1)} \land \neg (x \neq y) \land T_{1\rightarrow 1} \land x' \neq y') \times$

$SAT(F_{(0,l_0)} \land T_{l_0\rightarrow 1} \land x' \neq y')$
Example

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$l_0$</th>
<th>$1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

CTI $(1, x \neq y)$, level 1

$SAT(F_{(0,1)} \land \neg(x \neq y) \land T_{1 \rightarrow 1} \land x' \neq y') \times$

$SAT(F_{(0,l_0)} \land T_{l_0 \rightarrow 1} \land x' \neq y') \times$
Example

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>i</th>
<th>l:</th>
<th>l₀</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>$x = y$</td>
<td></td>
</tr>
</tbody>
</table>

CTI $(1, x \neq y)$, level 1

$SAT(F_{(0,1)} \land \neg(x \neq y) \land T_{1 \rightarrow 1} \land x' \neq y') \times$

$SAT(F_{(0,l₀)} \land T_{l₀ \rightarrow 1} \land x' \neq y') \times$
Example

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$l_0$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>$x = y$</td>
</tr>
</tbody>
</table>
IC3 on Control Flow Automata

Example

Frames $F_{(i,l)}$

<table>
<thead>
<tr>
<th>$i:$</th>
<th>$l$:</th>
<th>$l_0$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td></td>
<td>$x = y$</td>
</tr>
<tr>
<td>2</td>
<td>true</td>
<td></td>
<td>$x = y$</td>
</tr>
</tbody>
</table>

$\begin{align*}
x &++;
y &++;
x &\neq y
\end{align*}$
IC3 on Control Flow Automata

Evaluation

28 benchmarks from SVCOMP & device drivers, subset of [CG12].

![Graph showing the comparison between IC3SMT and IC3CFA](attachment:image.png)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>solved</th>
<th>solve time</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC3SMT</td>
<td>13/28</td>
<td>6328s</td>
<td>24328s</td>
</tr>
<tr>
<td>IC3CFA</td>
<td>22/28</td>
<td>584s</td>
<td>7784s</td>
</tr>
</tbody>
</table>

IC3 on Control Flow Automata

Evaluation

28 benchmarks from SVCOMP & device drivers, subset of [CG12].

![Graph showing solver performance](image)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>solved</th>
<th>solve time</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeIC3</td>
<td>21/28</td>
<td>1752s</td>
<td>10152s</td>
</tr>
<tr>
<td>IC3CFA</td>
<td>22/28</td>
<td>584s</td>
<td>7784s</td>
</tr>
</tbody>
</table>

IC3 on Control Flow Automata

Evaluation

28 benchmarks from SVCOMP & device drivers, subset of [CG12].

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>solved</th>
<th>solve time</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeIC3-ITP</td>
<td>28/28</td>
<td>3107s</td>
<td>3107s</td>
</tr>
<tr>
<td>IC3CFA</td>
<td>22/28</td>
<td>584s</td>
<td>7784s</td>
</tr>
</tbody>
</table>

Conclusion

Outline

Introduction

Preliminaries

Original IC3

Related Work

IC3 on Control Flow Automata

Conclusion
Conclusion

Contributions

Small SMT queries
Through inspection of only specific transitions, we can use a single edge formula instead of giving the whole transition relation to the solver.

No unrolling
By using $F_i$ frames in every location of the CFA, we can operate on the CFA exclusively. Thus no need for unrolling the CFA.

Stronger relative inductivity
When considering self-loops we can use the stronger relative inductivity that is used in the original IC3.
Conclusion

References


Aaron R. Bradley. “SAT-Based Model Checking without Unrolling”. In: VMCAI. 2011, pp. 70–87.

Alessandro Cimatti and Alberto Griggio. “Software Model Checking via IC3”. In: CAV. 2012, pp. 277–293.


Tobias Welp and Andreas Kuehlmann. “QF BV model checking with property directed reachability”. In: DATE. 2013, pp. 791–796.