# Better Generalization in IC3

Zyad Hassan Aaron R. Bradley Fabio Somenzi

Department of Electrical, Computer, and Energy Engineering
University of Colorado at Boulder

Oct 23, 2013

# Outline

1. Problem
2. Solution
3. Results
4. Analysis
5. Conclusions

# Outline

1. Problem
2. Solution
3. Results
4. Analysis
5. Conclusions

# IC3 [Bradley 2010,2011]

- Model checking algorithm for invariance properties
- Attempts to construct an inductive strengthening of the property
- Construction is incremental: derives many simple lemmas
- Lemmas generation either:
    - Results in an inductive strengthening
    - Guides the search to a counterexample trace
- SAT-based: performs many relatively easy SAT queries

# Generalization

- Key component of IC3
- Lifts IC3 from explicit to symbolic
- More successful generalization ⇔ Fewer individual states examined

*What does IC3 generalize?*

# Generalization

- Key component of IC3
- Lifts IC3 from explicit to symbolic
- More successful generalization ⇔ Fewer individual states examined

*What does IC3 generalize?*

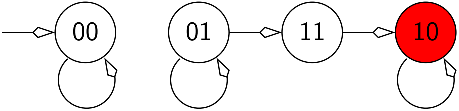
## Overview of IC3

- Prove the property by induction:
  - All initial states satisfy the property
  - All successors of good states are good

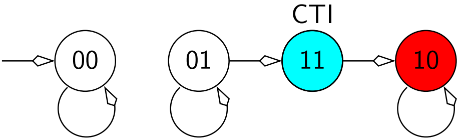
## Overview of IC3

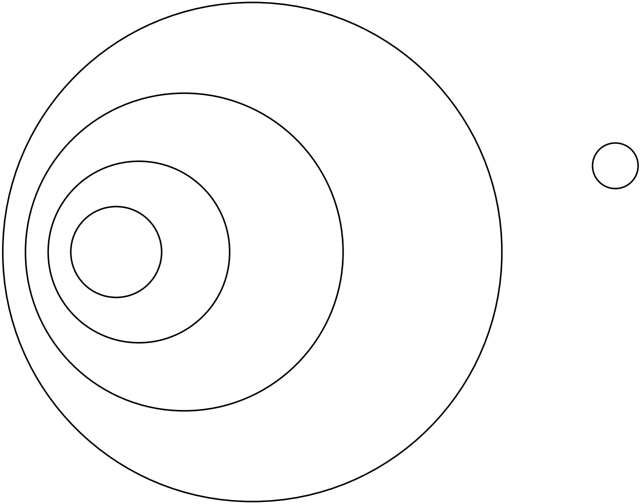
- Prove the property by induction:
  - All initial states satisfy the property
  - All successors of good states are good

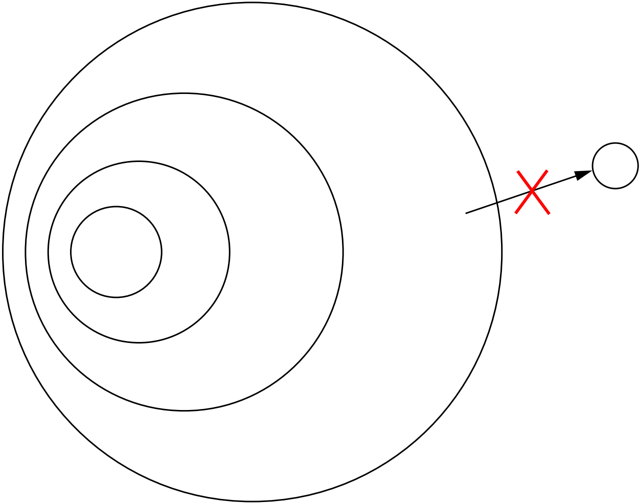
# Counterexamples to Induction (CTIs): The Troublemakers

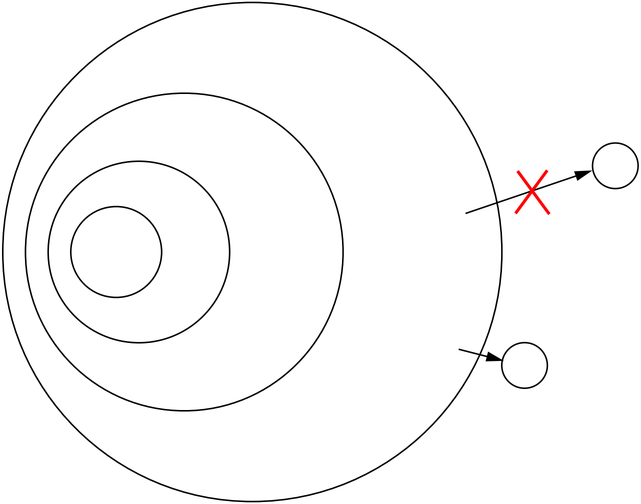


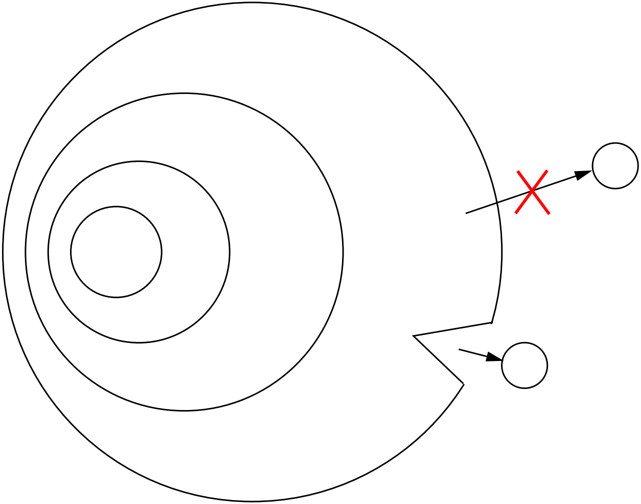
# Counterexamples to Induction (CTIs): The Troublemakers











# What does IC3 generalize?

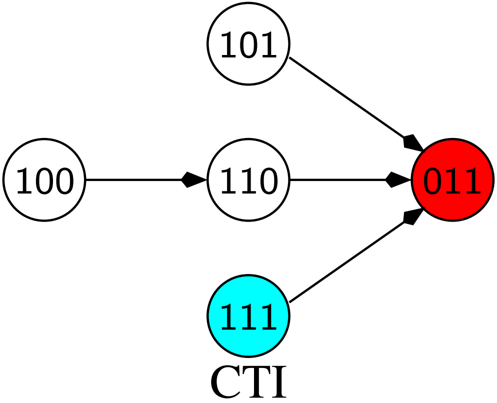
A state is unreachable within $k$ steps
to
A set of states is unreachable within $k$ steps

# How does generalization work?

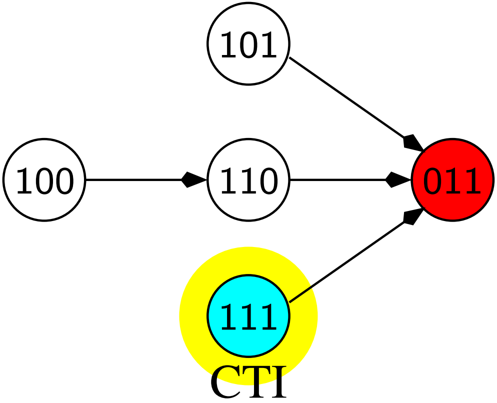
For each state-bit:

- Drop bit
- Find the smallest superset of states that have no predecessors outside of it (if exists)

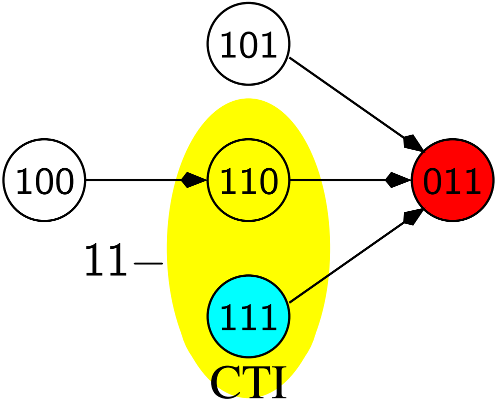
# Successful Generalization



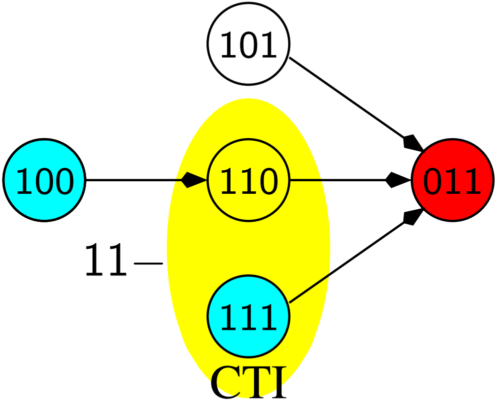
# Successful Generalization



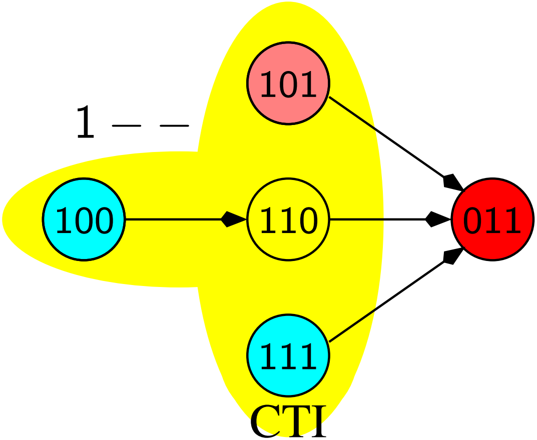
# Successful Generalization



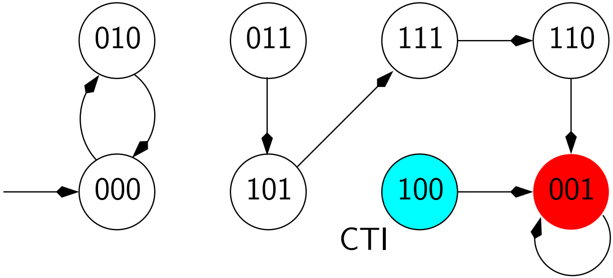
# Successful Generalization



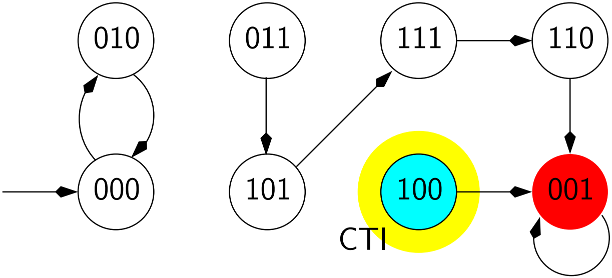
# Successful Generalization



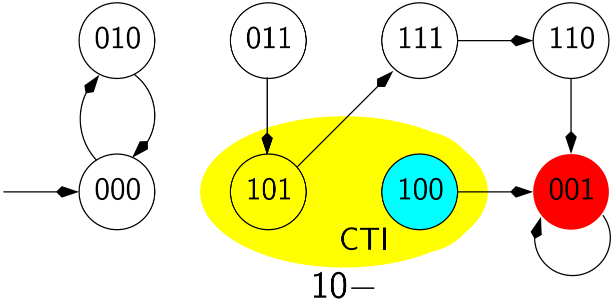
# Failed Generalization



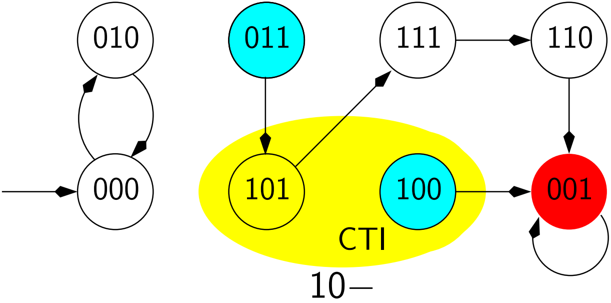
# Failed Generalization



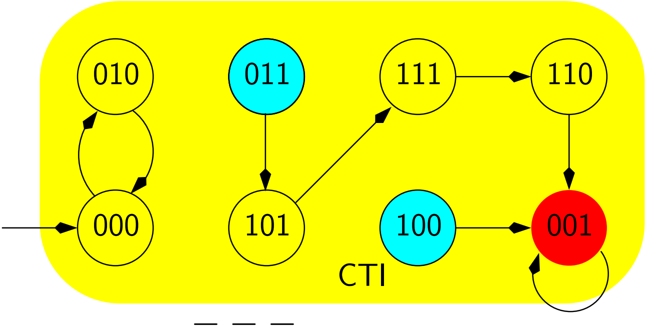
# Failed Generalization



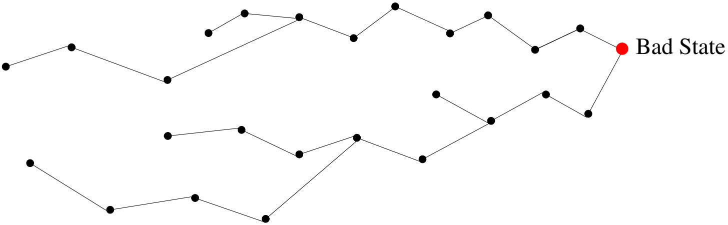
# Failed Generalization



# Failed Generalization



# Ineffective Generalization



# Outline

1 Problem

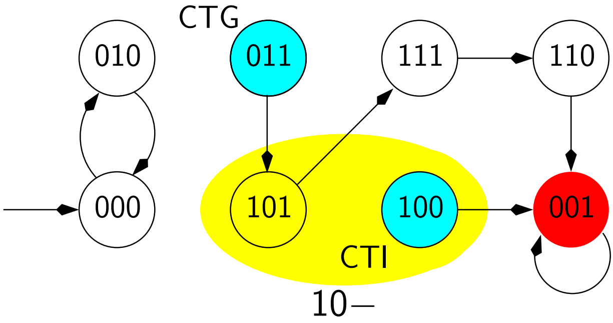
2 Solution

3 Results

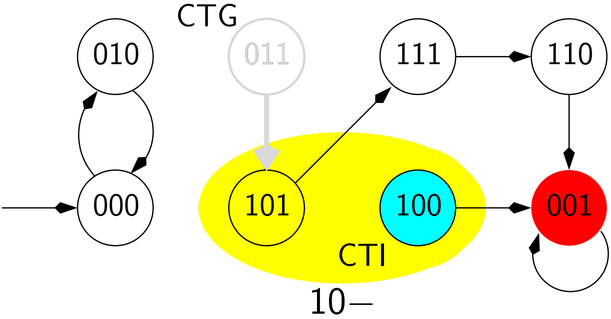
4 Analysis

5 Conclusions

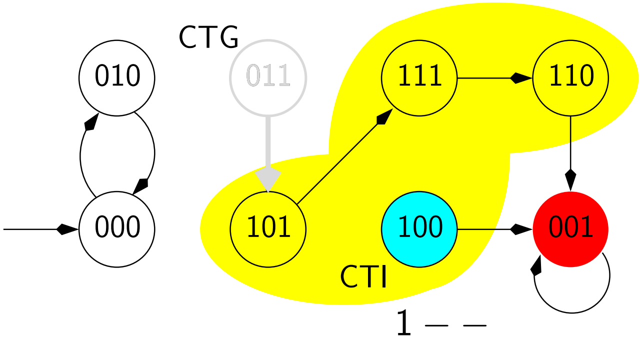
# Counterexamples to Generalization (CTGs)



# Counterexamples to Generalization (CTGs)



# Counterexamples to Generalization (CTGs)



# Counterexamples to Generalization (CTG)

- State preventing some generalization (dropping a specific state-bit)
- Unlike CTIs, not necessarily backward reachable
- Blocking CTGs:
    - Backward reachable: if deep, saves IC3 explicit traversal
    - Neither forward nor backward: never addressed by IC3 but could continue to obstruct generalization

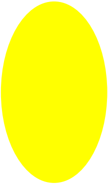
# ctgDown

- Instead of joining CTG with cube, turn attention to CTG
- Like CTIs, prove unreachable within $k$ steps
- If successful: generalize CTG, re-attempt CTI generalization
- If failed: join

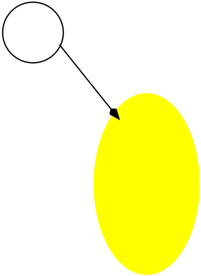
# ctgDown

- Instead of joining CTG with cube, turn attention to CTG if limit is not exceeded
- Like CTIs, prove unreachable within $k$ steps
- If successful: generalize CTG, re-attempt CTI generalization
- If failed: or exceeded maxCTGs limit, join, reset maxCTGs limit

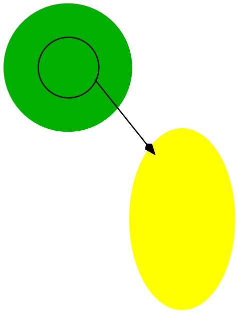
# Resetting Limit After Joins



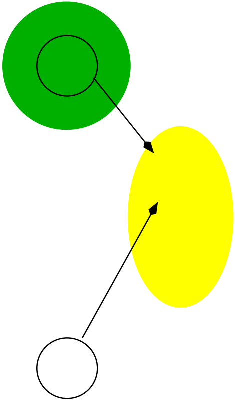
# Resetting Limit After Joins



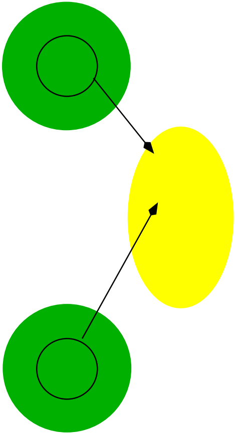
# Resetting Limit After Joins



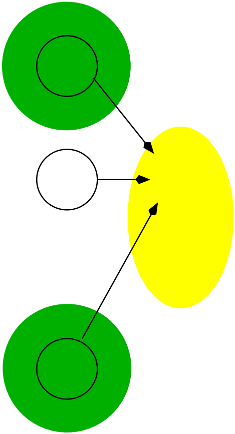
# Resetting Limit After Joins



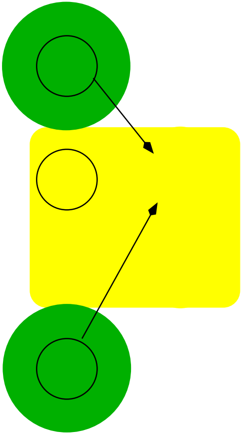
# Resetting Limit After Joins



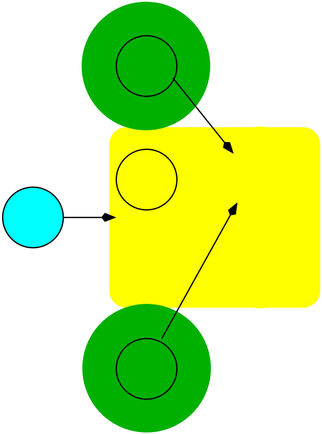
# Resetting Limit After Joins



# Resetting Limit After Joins



# Resetting Limit After Joins



# Outline

1. Problem
2. Solution
3. Results
4. Analysis
5. Conclusions

# Experimental Setup

- HWMCC'10+11+12 (beemb substituted by beemf)
- 900s timeout
- IImc and ABC
- Light-weight preprocessing
- 5 random seeds

# IImc

| | | Standard | | With ctgDown | | |
|---|---|---|---|---|---|---|
| Family | Size | Solved | Time (s) | Solved | Gain | Time (s) |
| 139 | 99 | 99 | 2524 | 99 | 0 | 1230 |
| 6s | 120 | 19 | 93466 | 21 | 2 | 94211 |
| beem | 86 | 48 | 38149 | 50 | 2 | 39594 |
| bob | 149 | 122 | 25804 | 120 | *(2)* | 28679 |
| intel | 60 | 23 | 35004 | 30 | 7 | 31153 |
| pdt | 350 | 331 | 19291 | 336 | 5 | 15469 |
| other | 280 | 271 | 11947 | 274 | 3 | 11463 |
| Total | 1144 | 913 | 226790 | 930 | **17** | 222460 |

# ABC

| | | Standard | | With ctgDown | | |
|---|---|---|---|---|---|---|
| Family | Size | Solved | Time (s) | Solved | Gain | Time (s) |
| 139 | 99 | 99 | 701 | 99 | 0 | 754 |
| 6s | 120 | 23 | 88401 | 30 | 7 | 82941 |
| beem | 86 | 51 | 34098 | 56 | 5 | 31191 |
| bob | 149 | 123 | 24292 | 124 | 1 | 24083 |
| intel | 60 | 23 | 35665 | 26 | 3 | 34249 |
| pdt | 350 | 329 | 22162 | 333 | 4 | 18120 |
| other | 280 | 270 | 12591 | 274 | 4 | 10359 |
| Total | 1144 | 916 | 218906 | 943 | **27** | 201417 |

# Outline

1 Problem

2 Solution

3 Results

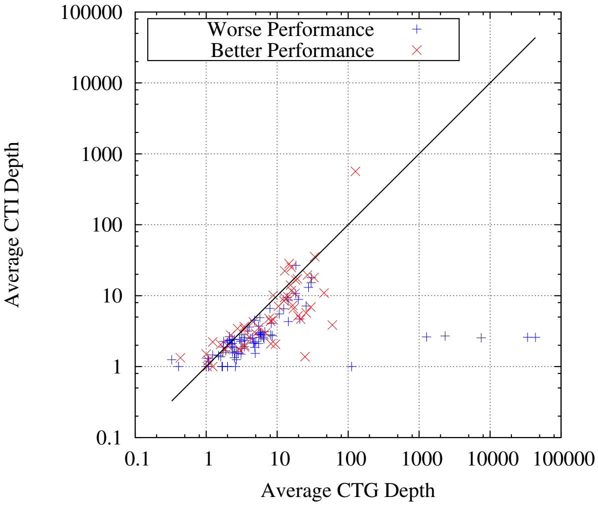
4 Analysis

5 Conclusions

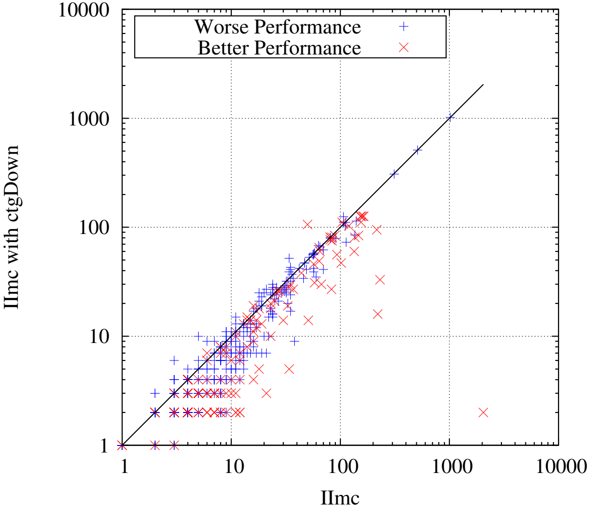
# Purpose

- Confirm reduction in length of explicit backward search
- Understand effect on various IC3 metrics

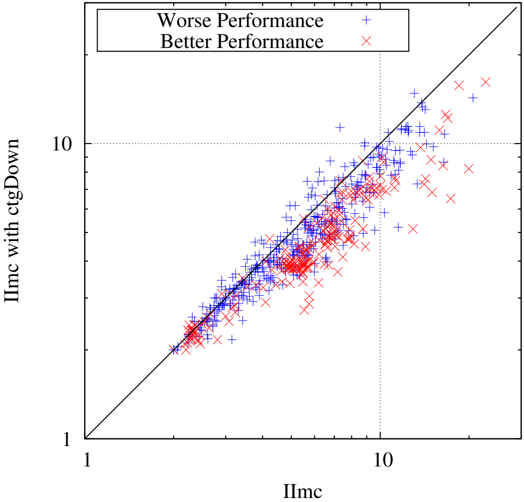
# Depth of CTGs vs. CTIs



# Effect on Maximum Depth of Priority Queue



# Effect on Average Clause Size



# Outline

1. Problem
2. Solution
3. Results
4. Analysis
5. Conclusions

# Conclusions

- Useful to divert IC3's attention to address reason for failure of generalization
- Not too aggressive handling of CTGs so as not to lose property focus
- Decreases depth of explicit search

# The End

Thank you.