

# **Symmetry Reduction and Heuristic Search for Error Detection in Model Checking**

*Workshop on Model Checking and Artificial Intelligence*  
*10. August 2003*

Alberto Lluch Lafuente? - Tilman Mehler?

lafuente@informatik.uni-freiburg.de

Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany

# Introduction

- Model checking as debugging tool.
- Counterexamples are used to fix errors.
- Main Drawbacks:
  - State Explosion Problem.
  - Long counterexamples difficult to understand.
- Heuristic search can:
  - Accelerate search for errors.
  - Obtain short counterexamples.
- Heuristic search successful only if combined with other techniques: partial order reduction, **symmetry reduction**.

# Model Checking Framework

- Asynchronous concurrent system.
- $n$  **identical** processes  $P_0..P_n$ .
- System's state space: LTS  $T = \langle S, s_0, \rightarrow, L \rangle$ 
  - $S$ : set of states
  - $s_0 \in S$  is the initial state
  - $\rightarrow \subseteq S \times S$ : transition relation
  - $L : S \rightarrow 2^{AP}$  labeling function.
- Verification of safety property.
- Bug finding phase: errors are common
- Explicit-state model checking.

# Symmetry Reduction

- Symmetries can be exploited to reduce the state space.
- Some properties are invariant under the symmetry.
- System remains behaviorally equivalent under symmetry permutations.
- Types of symmetries: rotational, full, mirror, etc.
- Detecting symmetries is difficult.
- Practical approaches: symmetric data types with restricted operations.
- Our assumption: correct symmetry is given.

# Definitions

- Symmetry relation  $\sim$  on  $S$ .
- $s_1, s_2 \in S$  symmetric iff  $s_1 \sim s_2$ .
- $s_1 \rightarrow s'_1, s_2 \rightarrow s'_2$  are *symmetric* iff  $s_1 \sim s_2$  and  $s'_1 \sim s'_2$ .
- $AP$  invariant under  $\sim$  iff  $s_1, \sim s_2 \sim L(s_1) = L(s_2)$ .
- $T_{/\sim} = \langle S_{/\sim}, [s_0], \Rightarrow, L_{/\sim} \rangle$ 
  - $[s]$ : *orbit* or equivalence class of  $s$ .
  - $[s_1] \Rightarrow [s'_1]$  if  $s_2 \rightarrow s'_2 \in \rightarrow$  such that  $s_2 \in [s_1]$  and  $s'_2 \in [s'_1]$ .
- If  $AP$  invariant under  $\sim$ ,  $L_{/\sim}([s]) = L(s')$  for some  $s' \in [s]$ .

# Orbit Problem

- $T/\sim$  analyzed by exploring a *representative* part of  $T$ .
- Orbit problem:  $s \sim s'$ ?
- Use representative states for  $[s]$ :  $rep : S \rightarrow S$ .
- $rep$  canonical iif  $s \sim s' \in S \Leftrightarrow rep(s) = rep(s')$ .
  - Lead to optimal reductions.
  - Computation can be time-expensive.
- Non unique representatives: normalizing functions
  - Lead to sub-optimal reductions.
  - Less computation time.

# Reachability and Bisimulation Equivalence

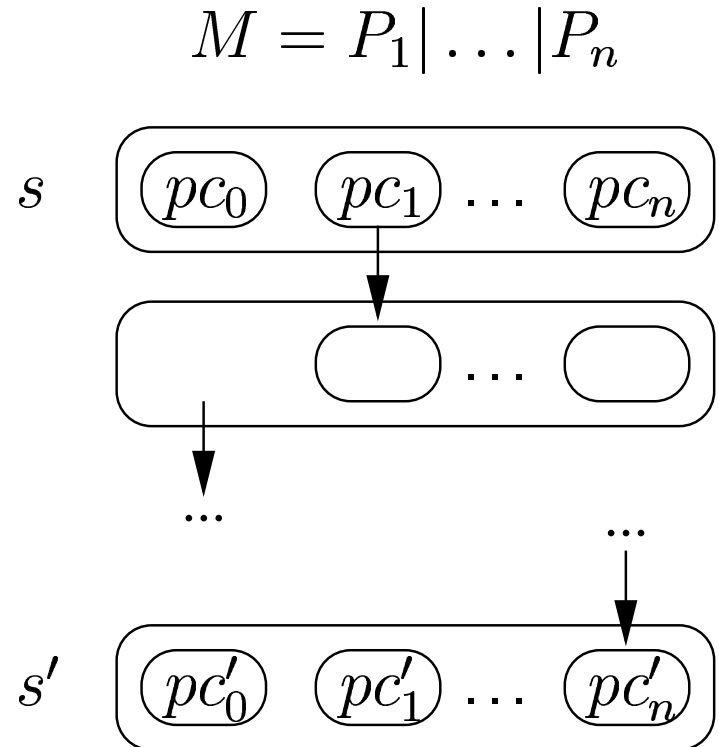
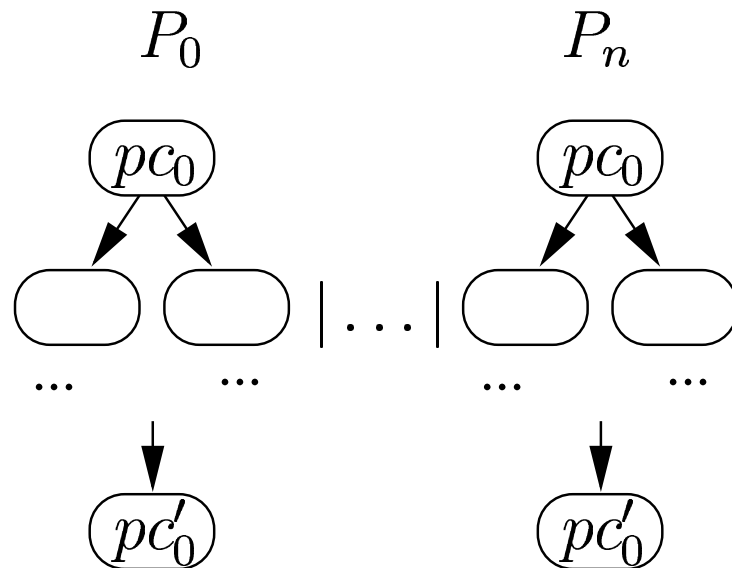
- A symmetry relation  $\sim$  preserves reachability:
  - if  $s_1 \rightsquigarrow s'_1$  and  $s_1 \sim s_2$  and  $\exists s'_2 \mid s'_2 \sim s'_1$  then  $s_2 \rightsquigarrow s'_2$ .
  - $s_1 \rightsquigarrow s'_1$  and  $s_2 \rightsquigarrow s'_2$  are symmetric paths.
- A bisimulation relation  $\sim$  preserves safety properties:
  - If  $s_1 \rightsquigarrow s'_1$  error path then every symmetric path is error path.
  - Safety verification reduced to reachability.
  - If  $e$  error state, then every state in  $[e]$  is error state.

# Heuristic Search in Model Checking

- Safety error detection reduced to reachability.
- Algorithms: **depth-first search**, breadth-first search.
- Drawbacks: long counterexamples, large exploration.
- Alternative: Heuristic search.
- LTS as (weighted) state transition graph.
- Heuristics guide the search to:
  - Accelerate the search.
  - Provide minimal counterexamples.
- Our approach:
  1. Best-First to find an error quickly.
  2. A\* to find minimal counterexample.

# FSM Distance Heuristic (1)

- From system state  $s = (pc_0, \dots, pc_n, v_0, \dots)$  to system state  $s' = (pc'_0, \dots, pc'_n, v'_0, \dots)$  each  $P_i$  must progress from  $pc_i$  to  $pc'_i$ .



# FSM Distance Heuristic (2)

- The minimal number of system transitions from  $s$  to  $s'$  is less or equal to the sum of the minimal distances from  $pc_i$  to  $pc'_i$  in each  $P_i$ .
- FSM Distance is a lower bound to the distance to  $s'$ .
- A\* is able to deliver the minimal path to  $s'$ .
- FSM distance is computed in  $O(n)$ .
  - Pre-computing the distances in  $P_i$  in  $O(|P_i|^3)$ .
  - In practice:  $|P_i| \ll |M|$ , since  $|M|$  is  $O(|P_1| \cdot \dots \cdot |P_n|)$ .

# Symmetry reduction and Heuristic Search

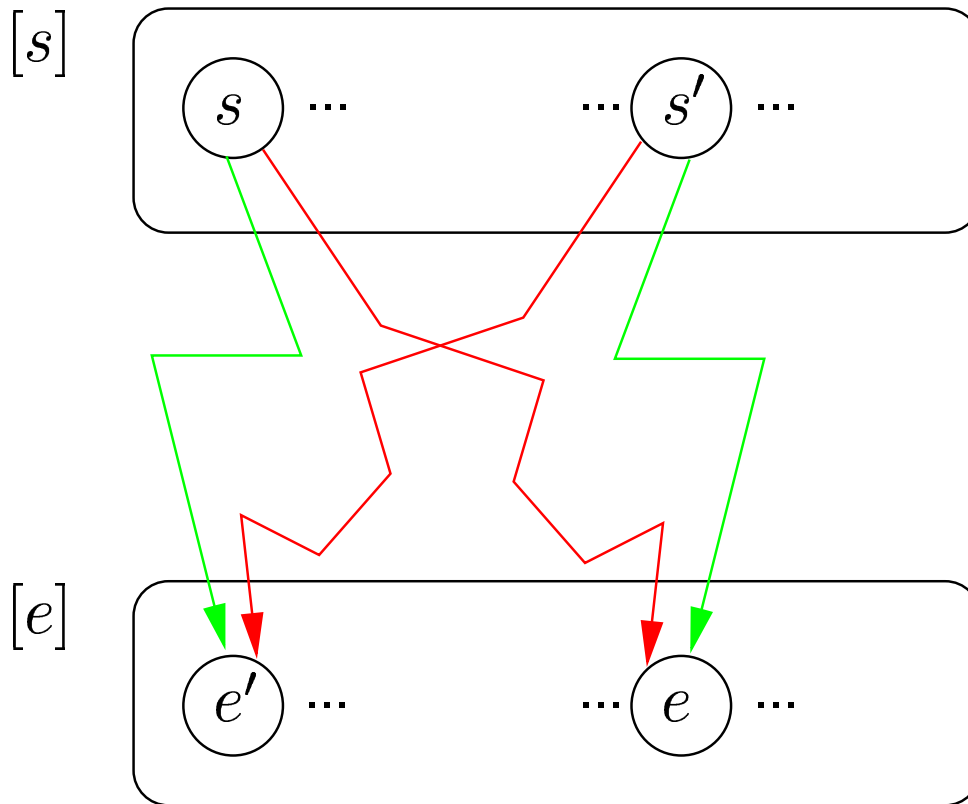
- Algorithm: straightforward combination.
- Symmetric paths have the same length.
- Symmetric paths have the same cost if symmetric transitions have the same cost.
- Heuristic  $h$  is symmetric iff  $s \sim s' \rightarrow h(s) = h(s')$ .
- Experiments with Best-first search
  - Combination better than one technique in isolation.
  - Symmetry reduction depends on kind of permutation.
  - Heuristic reduction depends on heuristic accuracy.

# Searching for an orbit

- Problem: find minimal path to error state  $e$ .
- “ $e$  never reached” not always invariant under  $\sim$ , but
  - $e$  violates  $f$  and  $f$  invariant under  $\sim$ .
  - “[ $e$ ] never reached” invariant under  $\sim$ .
- Find minimal path to  $e =$ 
  - Find minimal path to [ $e$ ]:  $s_0 \rightsquigarrow e', e' \in [e]$
  - Obtain symmetric path  $s_0 \rightsquigarrow e$ .
- How to check  $[s] = [e]$ ?
  - *rep* canonical:  $[s] = [e]$  iff  $rep(s) = rep(e)$
  - *rep* normalizing: store [ $e$ ] and make query.

# Distance to an orbit (1)

- $FD^e$  lower bound for the distance to  $e$  in  $T$ .
- $FD^e$  **not** lower bound for the distance to  $[e]$  in  $T/\sim$ .



minimal (symmetric) paths from:

$$\begin{aligned} s' \text{ to } e &\equiv [s] \text{ to } [e] \\ s \text{ to } e' &\equiv [s] \text{ to } [e] \end{aligned}$$

minimal (symmetric) paths from:

$$\begin{aligned} s \text{ to } e \\ s' \text{ to } e' \end{aligned}$$

# Distance to an orbit (2)

- Given  $h^e$  that estimates distance from  $s$  to  $e$  in  $T$ .
- $h_{/\sim}^{[e]}(s) = \min_{e' \in [e]} \{h^{e'}(s)\}$ .
- If  $h^e$  is admissible, then  $h_{/\sim}^{[e]}$  is admissible.
- If  $h^e$  is monotone, then  $h_{/\sim}^{[e]}$  is monotone.
- Time complexity of  $h_{/\sim}^{[e]} = T(h^e) \times |[e]|$ .
  - With rotational pid symmetries  $|[e]|$  is  $O(n)$  :)
  - With full pid symmetries  $|[e]|$  is  $O(n!)$  :(

# Implementations of $FD_{\sim}^{[e]}$

- Alternative 1
  - Store states of  $[e]$  with distinct pid permutations in table  $\{e\}$
  - Time complexity  $O(n \times |\{e\}|)$ .
  - $|\{e\}|$  is 1 if not pid permutations applied.
- Alternative 2 (current work)
  - Reduce to Minimum Weight Assignment Problem.
  - Time complexity  $O(n^3)$ .
- Alternative 2 better if  $O(|[e]|) > n^2$

# Experiments: Rotational

Leader election

	n	8	9	10
$FD^e$	states	2,106	2,641	3,117
	time	1.6	2.5	3.6
$FD_{/\sim}^{[e]}$	s	2,106	2,641	3,117
	time(I1)	1.6	2.5	3.6
	time(I2)	2.7	4.4	6.5
BFS	states	188,514	632,389	o.m.
	time	9:38.7	43:46	o.m.
ADFS	states	o.m.	o.m.	o.m.
	time	o.m.	o.m.	o.m.

# Experiments: Peterson

	n	4	5	6
$FD^e$	states	6,292	34,268	241,370
	time	2.8	22.7	8:07
$FD_{/\sim}^{[e]}$	states	5,134	39,885	455,634
	time(l1)	4.5	49.8	50:52
	time(l2)	4.67	1:0.1	30:20

n=6	$FD^e$	$FD1_{/\sim}^{[e]}$	$FD2_{/\sim}^{[e]}$
average time	$4.170e^{-6}$	$1.322e^{-3}$	$2.432e^{-4}$

# Experiments: Database

	n	6	7	8
$FD^e$	states	190	689	1,225
	time	0.4	7.9	1:48
$FD_{/\sim}^{[e]}$	s	343	499	1,524
	time(l1)	0.7	7.4	2:44
	time(l2)	1.0	8.4	2:59

	$FD^e$	$FD1_{/\sim}^{[e]}$	$FD2_{/\sim}^{[e]}$
Database (n=8)	$2.514e^{-6}$	$2.580e^{-4}$	$5.242e^{-4}$

# Experiments: Database (non cyclic)

	n	6	7	8
$FD^e$	states	398	391	1,978
	time	1.4	9.5	6:47
$FD_{/\sim}^{[e]}$	s	38	48	59
	time(l1)	0.2	1.3	12.1
	time(l2)	0.3	1.5	12.5

	$FD^e$	$FD1_{/\sim}^{[e]}$	$FD2_{/\sim}^{[e]}$
Database (n=8)	$2.514e^{-6}$	$2.580e^{-4}$	$5.242e^{-4}$

# Conclusions and Future Work

- Symmetry reduction and Heuristic search compatible.
  - Orthogonal combination.
- Finding minimal counterexample is important.
  - More efficient with canonical representatives.
  - Admissible heuristic  $FD_{/\sim}^{[e]}$ : two implementations.
  - Heuristic search outperforms blind search
- Current and Future Work: Experiments with Partial Order Reduction, Heuristic Search and Abstraction.