Adding Symmetry Reduction to UPPAAL

M. Hendriks\textsuperscript{1} G. Behrmann\textsuperscript{2} K.G. Larsen\textsuperscript{2}
P. Niebert\textsuperscript{3} F. Vaandrager\textsuperscript{1}

\textsuperscript{1}University of Nijmegen, The Netherlands
\textsuperscript{2}Aalborg University, Denmark
\textsuperscript{3}Université de Provence, France
Introduction

Motivation

- Exploitation of full symmetry can give factorial gain
- Full symmetry occurs in many timed systems
  - Fischer’s mutex protocol, CSMA/CD protocol ($\text{UPPAAL}$ benchmarks)
  - Dynamic configuration IPv4 addresses (Zhang & Vaandrager)
  - Distributed agreement algorithm (Attiya, Dwork, Lynch & Stockmeyer)
Introduction

Motivation

- Exploitation of full symmetry can give factorial gain
- Full symmetry occurs in many timed systems
  - Fischer’s mutex protocol, CSMA/CD protocol (Uppaal benchmarks)
  - Dynamic configuration IPv4 addresses (Zhang & Vaandrager)
  - Distributed agreement algorithm (Attiya, Dwork, Lynch & Stockmeyer)

Approach

  - Scalarsets as fully symmetric data type in description language
- Successfully used in several model checkers
  - Murϕ, Spin, Smv
Outline

(1) Some theory (Ip & Dill, 1993)

(2) Implementation
  • UPPAAL language enhancement
  • Representative computation

(3) Results

(4) Conclusions
Theory (Ip & Dill, 1993)

Syntactical level: system description

\[\text{P0} \quad \text{P1}\]

\[\text{A} \quad \text{A}\]

\[\text{B} \quad \text{B}\]

\[\text{C} \quad \text{C}\]
Theory (Ip & Dill, 1993)

Syntactical level: system description

Semantical level: state graph

\((Q, Q_0, \Delta)\)
Theory (Ip & Dill, 1993)

Syntactical level: system description

Semantical level: state graph

\[(Q, Q_0, \Delta)\]

Detect bijections \(h : Q \rightarrow Q\) in state graph from system description such that

\[\begin{align*}
&\triangleq q \in Q_0 \iff h(q) \in Q_0 \\
&\triangleq (q_1, q_2) \in \Delta \iff (h(q_1), h(q_2)) \in \Delta
\end{align*}\]
Theory (2)

Automorphism $h$ on state graph $G$
Theory (2)

Automorphism $h$ on state graph $G$ 

$h$ induces *quotient graph* $G'$
Theory (2)

Automorphism $h$ on state graph $G$ \hspace{1cm} $h$ induces quotient graph $G'$

Then: $q$ reachable in $G$ $\iff$ $[q]$ reachable in $G'$
Implementation

(1) Find a set of automorphisms $H$ from the system description
   • Introduce a symmetric data type, e.g., scalarsets

(2) During state space exploration: $[q] = ? [q']$ (orbit problem)
   • Use a representative function $\theta : Q \rightarrow Q$
Language enhancements

Template header:
process F (const proc_id pid)

Local declarations:
clock x;

Global declarations:
typedef scalarset[3] proc_id;
proc_id id;
bool set;

Process assignments:
Procs = forall i in proc_id : F(i);

System description:

Adding Symmetry Reduction to UPPAAL – FORMATS 2003, September 6-7 2003, Marseille, France
Swap process 0 with process 1
State swap example

Swap process 0 with process 1
State swap example (2)

Swap process 1 with process 2
State swap example (2)

Swap process 1 with process 2
Representative computation

Idea: “minimize” state using state swaps w.r.t. some total order

Problem: symbolic representation of sets of clock valuations (zones)

Solution: diagonal property of zones
Diagonal property

Let \( x \) and \( y \) be clocks and let \( Z \) be a zone (set of clock valuations)

\[
x \leq_{Z} y \iff \forall \nu \in Z \; \nu(x) \leq \nu(y)
\]

\[
x \approx_{Z} y \iff \forall \nu \in Z \; \nu(x) = \nu(y)
\]

\[
x \prec_{Z} y \iff (x \leq_{Z} y \land x \not\approx_{Z} y)
\]

Lemma (diagonal property): Consider a symbolic forward state space exploration algorithm. Assume that the clocks are reset to the value 0 only. For all states \((\vec{l}, v, Z)\) stored in the waiting and passed list and for all clocks \(x\) and \(y\) holds that either \(x \prec_{Z} y\), \(y \prec_{Z} x\), or \(x \approx_{Z} y\).
Diagonal property: proof sketch

1. Initial zone satisfies diagonal property (all clocks equal 0)
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset

(3) Time elapse
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset

(3) Time elapse
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset

(3) Time elapse

(4) Intersection
Diagonal property: proof sketch

(1) Initial zone satisfies diagonal property (all clocks equal 0)

(2) Clock reset

(3) Time elapse

(4) Intersection
Representative computation (2)

Diagonal property gives a total order on clocks (and on states)
  • Easily decidable using the DBM representation of zones

State swaps implement transpositions of scalarset elements
  • All permutations of scalarset elements can be obtained

Representative computation by minimization of state
  • “Bubble sort” the state with state swaps w.r.t. the total order
  • Canonical under certain assumptions that involve the discrete part of the state
Results

Adding Symmetry Reduction to UPPAAL – FORMATS 2003, September 6-7 2003, Marseille, France
Conclusions