Bisimulation Minimisations for Boolean Equation Systems

Jeroen J.A. Keiren Tim A.C. Willemse

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven

Model Checking

μ-Calculus model checking problem: answer $L \models f$

- L is a Labelled Transition System;
- f is a μ-calculus formula;

- Model checking problem $=$ solving Boolean equation systems
- A BES is a sequence of fixed point equations
- Size BES $\mathcal{E}$ encoding $L \models f$ ..... $\mathcal{O}(|L| \cdot |f|)$
- Solving BES $=$ finding the winner in Parity Games
- Algorithm for computing winner PG $=$ algorithm solving BES

- Solving BES/Computing winners in PG:
  - Small Progress Measures [Jurdziński'00]
  - Bigstep [Schewe'07]
  - Strategy Improvement [Jurdziński & Vöge'00]
- All are exponential, and problem is in NP $\cap$ co-NP

Contributions of this work

- Logical equivalences for BESs;
- Identified reductions for practical examples;
- Identified strengths compared to reductions on LTSs.

Boolean equation system [Mader'97]

Definition (Boolean Equation System (BES))

A BES $\mathcal{E}$ in SRF is defined by the following grammar:

$$\begin{aligned} \mathcal{E} &::= \epsilon \mid (\mu X = f)\ \mathcal{E} \mid (\nu X = f)\ \mathcal{E} \\ f &::= X \mid \bigvee F \mid \bigwedge F \end{aligned}$$

- ϵ is the empty BES;
- μ, ν are least, resp., greatest fixed point signs;
- X is a proposition variable from a set $\mathcal{X}$;
- F is a non-empty set of proposition variables.

Closed BES

- $\mathrm{bnd}(\mathcal{E})$ bound variables (occurring in left hand side of an equation);
- $\mathrm{occ}(\mathcal{E})$ occurring variables (occurring in right hand side of an equation);
- $\mathcal{E}$ is closed: $\mathrm{occ}(\mathcal{E}) \subseteq \mathrm{bnd}(\mathcal{E})$;
- $\mathcal{B}$ is the set of closed BES in SRF;
- $X \trianglelefteq Y$: the equation for X precedes Y's in $\mathcal{E}$;

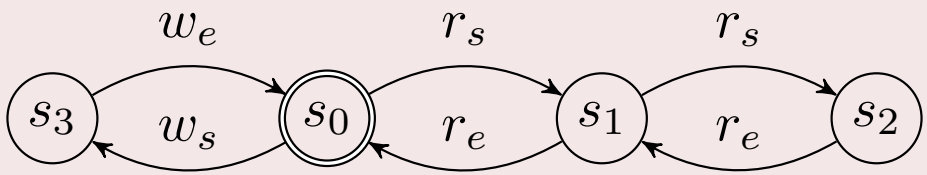
Solution

- Solution satisfies Boolean equations ignoring fixpoint symbols;
- Specific solution given by fixpoint symbols;
- Intuitively: left-most fixpoint symbols have highest priority;
- Solution is order-sensitive:

$$\underset{X = Y = \text{false}}{[\![(\mu X = Y)\ (\nu Y = X)]\!]} \quad \neq \quad \underset{X = Y = \text{true}}{[\![(\nu Y = X)\ (\mu X = Y)]\!]}$$

Academic example BES

Example (Mutual exclusion)



On some path a reader can infinitely often start reading:

$$\nu X.\mu Y.\ \langle r_s \rangle X \vee \langle \overline{r_s} \rangle Y$$

Corresponding BES (translation of [Mader'97]):

$$(\nu X_{s_0} = Y_{s_0})\ (\nu X_{s_1} = Y_{s_1})\ (\nu X_{s_2} = Y_{s_2})\ (\nu X_{s_3} = Y_{s_3})$$
$$(\mu Y_{s_0} = X_{s_1} \vee Y_{s_1})\ (\mu Y_{s_1} = X_{s_2} \vee Y_{s_0})\ (\mu Y_{s_2} = Y_{s_1})\ (\mu Y_{s_3} = Y_{s_0})$$

Rank and Op

- rank(X) indicates in which block of like-signed equations X occurs;
- rank(X) is odd iff X is defined in a μ-equation;
- rank(X) inductively defined on structure of BES;
- op(X) indicates top-level boolean operator of equation for X.

Example (Rank/Op)

rank(_)	op(_)			
(1)	$\wedge$	μX	$=$	$X \wedge (Y \vee Z)$
(2)	$\vee$	νY	$=$	$W \vee (X \wedge Y)$
(3)	$\bot$	μZ	$=$	Z
(3)	$\vee$	μW	$=$	$Z \vee (Z \vee W)$

Solution equivalence

Definition

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. We say equations for X and Y are solution equivalent, denoted $X \equiv Y$, if $[\![\mathcal{E}]\!](X) = [\![\mathcal{E}']\!](Y)$;
we say $\mathcal{E}$ and $\mathcal{E}'$ are solution equivalent, denoted $\mathcal{E} \equiv \mathcal{E}'$, if their first equations are solution equivalent.

Observations:

- $\equiv$ is the coarsest equivalence relation in possible lattice on BESs;
- Computing $\equiv$ is in NP $\cap$ co-NP.

Local equivalences for Boolean Equation Systems

Aim

Identify finer equivalence relation for BESs that:

- is efficiently (polynomial time) computable;
- preserves solution of BES;

Strong Bisimilarity

Definition

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. A relation $\mathcal{R} \subseteq \mathsf{bnd}(\mathcal{E}) \times \mathsf{bnd}(\mathcal{E}')$ is said to be a bisimulation if, whenever $X \mathcal{R} Y$, then:

- $\mathsf{rank}(X) = \mathsf{rank}(Y)$;
- $\mathsf{op}(X) = \mathsf{op}(Y)$;
- for all $U \in \mathsf{occ}(X)$, there is a $V \in \mathsf{occ}(Y)$, such that $U \mathcal{R} V$;
- for all $V \in \mathsf{occ}(Y)$, there is a $U \in \mathsf{occ}(X)$, such that $U \mathcal{R} V$;

Equations for X and Y are bisimilar, denoted $X \sim Y$, if there exists a bisimulation relation $\mathcal{R}$ such that $X \mathcal{R} Y$;

$\mathcal{E}$ and $\mathcal{E}'$ are bisimilar, denoted $\mathcal{E} \sim \mathcal{E}'$, if their first equations are bisimilar.

Properties

- Bisimilarity is an equivalence relation over $\mathcal{B}$;

Lemma

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. If $\mathcal{E}$ and $\mathcal{E}'$ are bisimilar, then they are also solution equivalent.

Theorem

The relation $\sim$ is strictly finer than $\equiv$.

Application

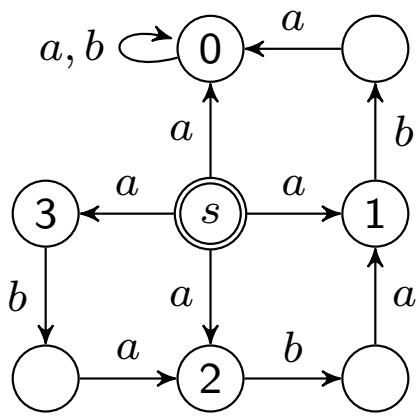
Example

Let N be an arbitrary positive natural number. Consider the process described by the following set of recursive processes (using process-algebraic notation in ACP-style):

$$\begin{aligned}\{ \quad & S = \sum\{a \cdot X(n) \mid n \leqslant N\}, \\ & X(0) = a.X(0) + b.X(0), \quad X(n+1) = b.a.X(n) \quad \}\end{aligned}$$

Application

Consider property $\phi = \nu Y.\langle a \rangle([a]\text{false} \wedge \nu Z.\langle b \rangle \langle a \rangle Z)$
Informally: after an a action, it is possible to see infinite trace $(ba)^\omega$



Visualisation of S for $N = 3$

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation system for $\mathcal{E}_3$

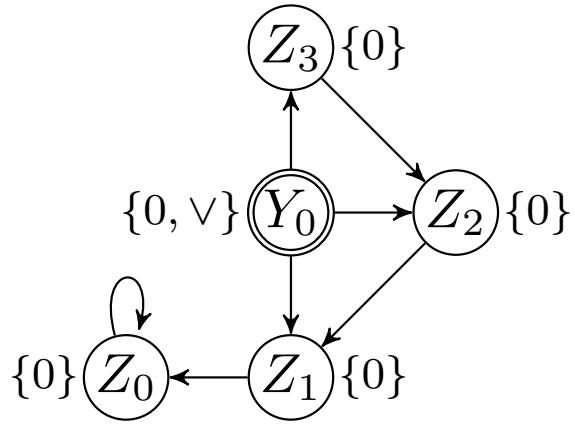
Observations:

- S consists of $2(N+1)$ states;
- S minimal modulo strong bisimilarity.

Application (Strong Bisimulation)

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation system for $\mathcal{E}_3$

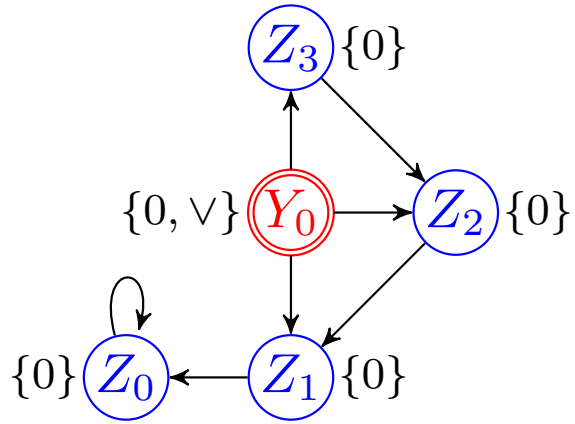


$\mathcal{E}_3$'s Dependency Graph

Application (Strong Bisimulation)

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation system for $\mathcal{E}_3$

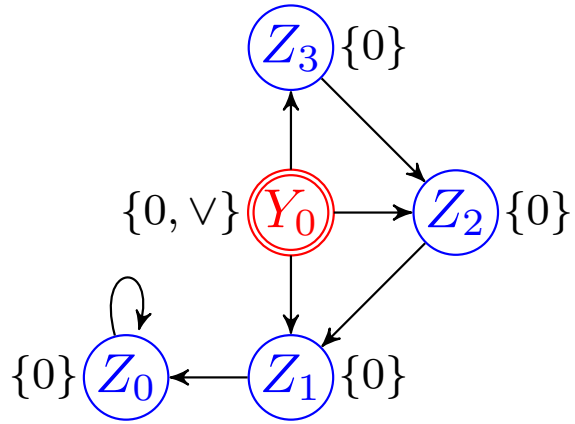


$\mathcal{E}_3$'s Dependency Graph

Application (Strong Bisimulation)

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

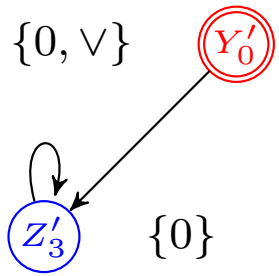
Equation system for $\mathcal{E}_3$



$\mathcal{E}_3$'s Dependency Graph

$$
\begin{aligned}
(\nu Y_{0/\sim} &= Z_{3/\sim} \vee Z_{3/\sim}) \\
(\nu Z_{3/\sim} &= Z_{3/\sim})
\end{aligned}
$$

Equation system for $\mathcal{E}_{3/\sim}$



$\mathcal{E}_{3/\sim}$'s Dependency Graph

Idempotence identifying Bisimilarity

Problem with strong bisimilarity:

- Equations $\sigma X = X' \wedge X''$, or $\sigma X = X' \vee X''$, with $X' \sim X''$ cannot be minimised to $\sigma X = X'$;
- Minimising to $\sigma X = X'$ nevertheless changes operand of equation, violating $\mathcal{E} \sim \mathcal{E}'$.

Aim

Define coarser equivalence, such that logical operand is only considered when applied to proposition variables from distinguishable classes.

Strong Bisimilarity: Definition

Definition

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. A relation $R \subseteq \mathsf{bnd}(\mathcal{E}) \times \mathsf{bnd}(\mathcal{E}')$ is said to be a strong bisimulation if, whenever $X \mathcal{R} Y$, then:

- $\mathsf{rank}(X) = \mathsf{rank}(Y)$;
- $\mathsf{op}(X) = \mathsf{op}(Y)$;
- for each $U \in \mathsf{occ}(X)$ there is a $V \in \mathsf{occ}(Y)$ such that $U \mathcal{R} V$;
- for each $V \in \mathsf{occ}(Y)$ there is a $U \in \mathsf{occ}(X)$ such that $U \mathcal{R} V$.

Equations for X and Y are strongly bisimilar, denoted $X \sim Y$, if there is a strong bisimulation relation R such that $X \mathcal{R} Y$;
$\mathcal{E}$ and $\mathcal{E}'$ are strongly bisimilar, denoted $\mathcal{E} \sim \mathcal{E}'$, if their first equations are strongly bisimilar.

Idempotence identifying Bisimilarity: Definition

Definition

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. A relation $R \subseteq \mathsf{bnd}(\mathcal{E}) \times \mathsf{bnd}(\mathcal{E}')$ is said to be an idempotence identifying bisimulation if, whenever $X \mathcal{R} Y$, then:

- $\mathsf{rank}(X) = \mathsf{rank}(Y)$;
- $\mathsf{op}(X) \neq \mathsf{op}(Y) \Rightarrow$ for all $U \in \mathsf{occ}(X)$ and $V \in \mathsf{occ}(Y)$: $U \mathcal{R} V$;
- for each $U \in \mathsf{occ}(X)$ there is a $V \in \mathsf{occ}(Y)$ such that $U \mathcal{R} V$;
- for each $V \in \mathsf{occ}(Y)$ there is a $U \in \mathsf{occ}(X)$ such that $U \mathcal{R} V$.

Equations for X and Y are idempotence identifying bisimilar, denoted $X \sim_{ii} Y$, if there is an idempotence identifying bisimulation relation R such that $X \mathcal{R} Y$;
$\mathcal{E}$ and $\mathcal{E}'$ are idempotence identifying bisimilar, denoted $\mathcal{E} \sim_{ii} \mathcal{E}'$, if their first equations are idempotence identifying bisimilar.

Idempotence identifying Bisimilarity: Properties

Lemma

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. If $\mathcal{E}$ and $\mathcal{E}'$ are idempotence identifying bisimilar, then they are also solution equivalent.

Theorem

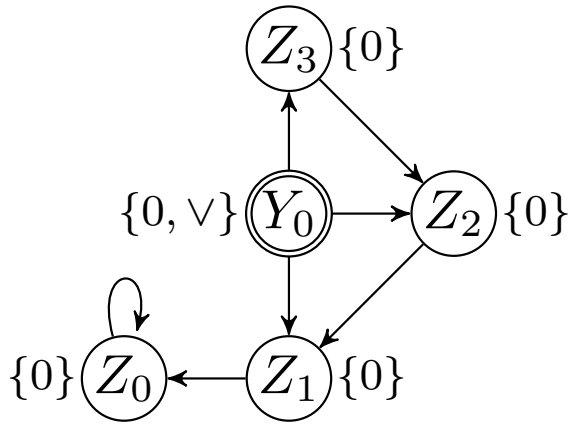
1. *the relation $\sim$ is strictly finer than $\sim_{ii}$;*
2. *the relation $\sim_{ii}$ is strictly finer than $\equiv$;*

Idempotence identifying Bisimilarity: Application

Recall our example:

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation System $\mathcal{E}_3$



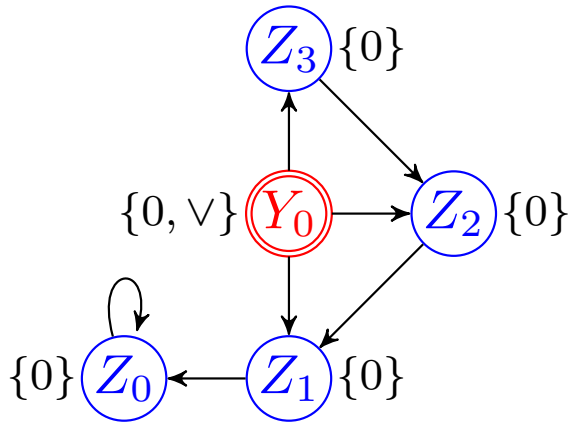
$\mathcal{E}_3$'s Dependency Graph

Idempotence identifying Bisimilarity: Application

Strong bisimulation:

$$
\begin{aligned}
&(\nu Y_0 &&= Z_3 \vee Z_2 \vee Z_1)\\
&(\nu Z_3 &&= Z_2)\\
&(\nu Z_2 &&= Z_1)\\
&(\nu Z_1 &&= Z_0)\\
&(\nu Z_0 &&= Z_0)
\end{aligned}
$$

Equation System $\mathcal{E}_3$



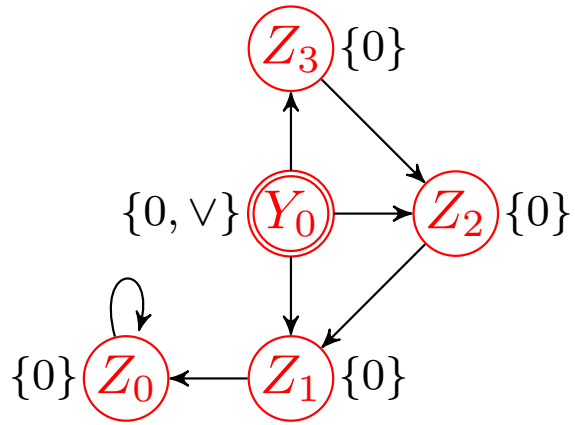
$\mathcal{E}_3$'s Dependency Graph

Idempotence identifying Bisimilarity: Application

Idempotence identifying bisimulation:

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation System $\mathcal{E}_3$



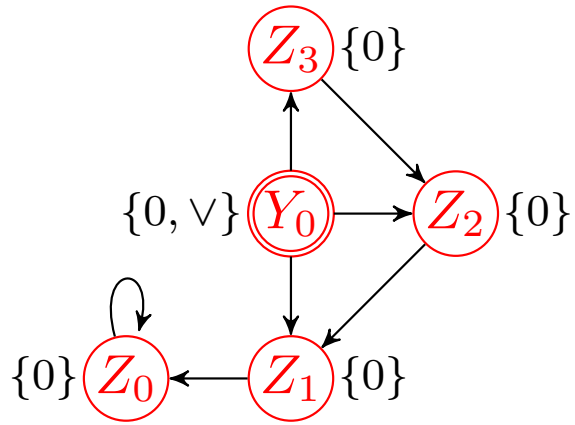
$\mathcal{E}_3$'s Dependency Graph

Idempotence identifying Bisimilarity: Application

Idempotence identifying bisimulation:

$$
\begin{aligned}
(\nu Y_0 &= Z_3 \vee Z_2 \vee Z_1) \\
(\nu Z_3 &= Z_2) \\
(\nu Z_2 &= Z_1) \\
(\nu Z_1 &= Z_0) \\
(\nu Z_0 &= Z_0)
\end{aligned}
$$

Equation System $\mathcal{E}_3$

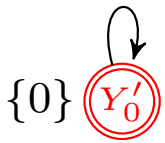


$\mathcal{E}_3$'s Dependency Graph

Reduced:

$$(\nu Y_{0/\sim_{ii}} = Y_{0/\sim_{ii}})$$

Equation system $\mathcal{E}_{3/\sim_{ii}}$



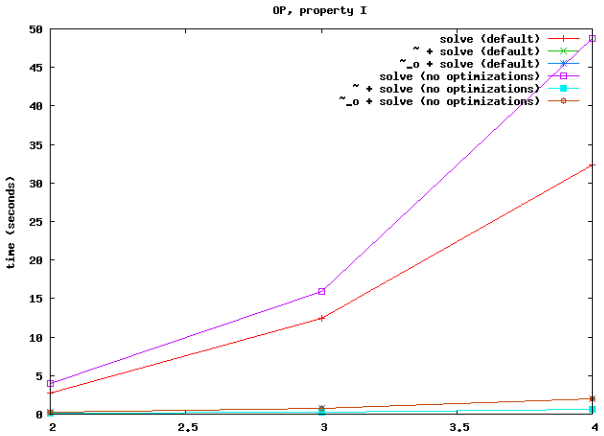
$\mathcal{E}_{3/\sim_{ii}}$'s dependency graph

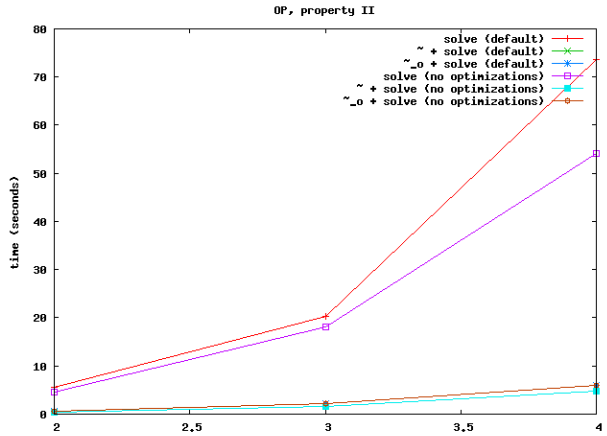
Observations

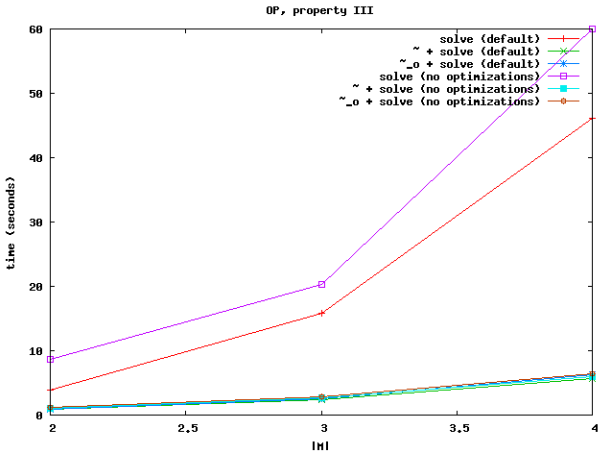
- Both equivalences reduce BES, whereas LTS could not be reduced;
- Idempotence identifying bisimilarity can reduce further than strong bisimilarity;
- Idempotence identifying bisimilarity can give rise to arbitrarily large reduction;
- Both equivalences computable in $\mathcal{O}(m \log n)$ time (*e.g.* Paige-Tarjan).

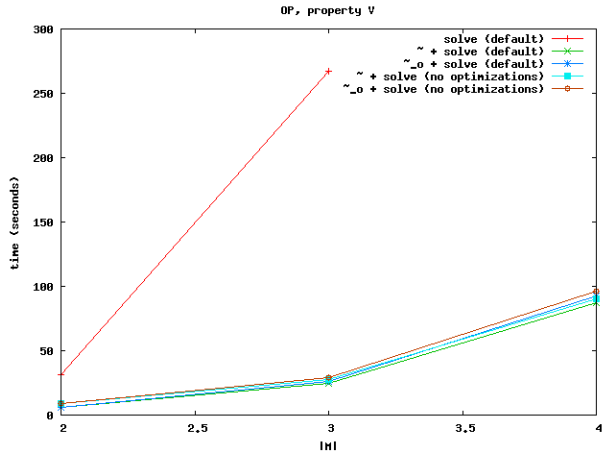
Experiments

Some experimental results (Onebit protocol with four properties):









Conclusions

- We have defined two equivalence relations on BES, *viz.* strong bisimulation and idempotence identifying bisimulation;
- Idempotence identifying bisimulation solves some peculiarities displayed by strong bisimulation;
- Both relations preserve solution equivalence;
- Both relations are efficiently computable ($\mathcal{O}(m \log n)$);
- Minimisation is rewarding in practice.

Related & Future work

Related work:

- Generalisation to arbitrary closed equation systems (Reniers & Willemse)

Future work:

- Extension to coarser notions of equivalence, *e.g.* stuttering equivalence (in progress);
- Extension to open systems of equations (in progress);
- Characterise relation between strong bisimulation on LTS and strong bisimulation on BES;
- Generalise to Parameterised Boolean Equation Systems.