

Bisimulation Minimisations for Boolean Equation Systems

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

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven

Introduction

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 solving BES = algorithm for computing winner PG

Introduction

- Solving BES/Computing winners in PG:
 - Small Prog. Meas. [Jurdziński'00] $\mathcal{O}(ah(\mathcal{E}) \cdot m \cdot (\frac{n}{2})^{\frac{ah(\mathcal{E})}{2}})$
 - Bigstep [Schewe'07] $\mathcal{O}(n \cdot m^{\frac{ah(\mathcal{E})}{3}})$
 - Strat. Impr. [Jurdziński & Vöge'00] $\mathcal{O}(2^{ah(\mathcal{E})} \cdot n \cdot m)$
- 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

Definition (Boolean Equation System (BES))

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

$$\mathcal{E} ::= \epsilon \mid (\mu X = f) \mathcal{E} \mid (\nu X = f) \mathcal{E}$$

$$f ::= X \mid \bigvee F \mid \bigwedge F$$

- 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 and well-formed BES

bound/occurring proposition variables

$$\text{bnd}(\epsilon) = \emptyset$$

$$\text{bnd}((\sigma X = f) \mathcal{E}) = \text{bnd}(\mathcal{E}) \cup \{X\}$$

$$\text{occ}(\epsilon) = \emptyset$$

$$\text{occ}((\sigma X = f) \mathcal{E}) = \text{occ}(\mathcal{E}) \cup \text{occ}(f)$$

$$\text{occ}(c) = \emptyset$$

$$\text{occ}(f \vee g) = \text{occ}(f) \cup \text{occ}(g)$$

$$\text{occ}(X) = \{X\}$$

$$\text{occ}(f \wedge g) = \text{occ}(f) \cup \text{occ}(g)$$

- \mathcal{E} is **closed**: $\text{occ}(\mathcal{E}) \subseteq \text{bnd}(\mathcal{E})$;
- \mathcal{B} is the set of closed BES in SRF;
- $X \triangleleft Y$: the equation for X precedes Y 's in \mathcal{E} ;
- \mathcal{E} is well-formed: \triangleleft is **acyclic**.

Interpretation

Proposition formulae are interpreted in the context of an **environment** $\eta: \mathcal{X} \rightarrow \mathbb{B}$, assigning Boolean values to proposition variables;

Definition (Interpretation of proposition formulae)

Let $\eta: \mathcal{X} \rightarrow \mathbb{B}$ be an environment. **Interpretation** $\llbracket f \rrbracket \eta$ maps a proposition formula f to true or false:

$$\begin{array}{ll} \llbracket c \rrbracket \eta = c & \llbracket X \rrbracket \eta = \eta(X) \\ \llbracket f \vee g \rrbracket \eta = \llbracket f \rrbracket \eta \vee \llbracket g \rrbracket \eta & \llbracket f \wedge g \rrbracket \eta = \llbracket f \rrbracket \eta \wedge \llbracket g \rrbracket \eta \end{array}$$

Solution

- **Solution** to BES defined in context of environment $\eta: \mathcal{X} \rightarrow \mathbb{B}$.
- Solution satisfies Boolean equations ignoring fixpoint symbols;
- Specific solution given by fixpoint symbols;
- Intuitively: left-most fixpoint symbols have highest priority;

Peculiarities:

- Solution to closed BES **independent** of environment:

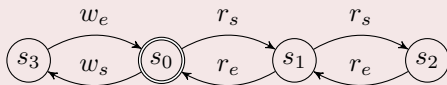
$$\forall X \in \text{bnd}(\mathcal{E}) : \llbracket \mathcal{E} \rrbracket \eta(X) = \llbracket \mathcal{E} \rrbracket \eta'(X) \text{ for all } \eta, \eta'$$

- Solution is **order-sensitive**:

$$\llbracket (\mu X = Y) (\nu Y = X) \rrbracket \neq \llbracket (\nu Y = X) (\mu X = Y) \rrbracket$$

Academic example BES

Example (Mutual exclusion)



On some path a reader can infinitely often start reading:

$$\nu X. \mu Y. \langle r_s \rangle X \vee \langle \bar{r}_s \rangle Y$$

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

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

Example (Rank/Op)

$\text{rank}(-)$	$\text{op}(-)$		
(1)	\wedge	μX	$= X \wedge (Y \vee Z)$
(2)	\vee	νY	$= W \vee (X \wedge Y)$
(3)	\perp	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 $\llbracket \mathcal{E} \rrbracket(X) = \llbracket \mathcal{E}' \rrbracket(Y)$;
we say \mathcal{E} and \mathcal{E}' are solution equivalent, denoted $\mathcal{E} \equiv \mathcal{E}'$, if their first equations are solution equivalent.

Observation:

- \equiv is the coarsest equivalence relation in possible lattice on BESs;

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 $\mathcal{R} \subseteq \text{bnd}(\mathcal{E}) \times \text{bnd}(\mathcal{E}')$ is said to be a **bisimulation** if, whenever $X\mathcal{R}Y$, then:

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

Quotienting

Definition

Let $\mathcal{E} \in \mathcal{B}$. The **quotient** of \mathcal{E} , denoted \mathcal{E}/\sim is an equation system consisting of equations $\sigma_i C_i = f_i$, for $i \in [1..n]$, where:

- $C_i \in \text{bnd}(\mathcal{E})/\sim$, i.e., each $C_i \subseteq \text{bnd}(\mathcal{E})$ is exactly one equivalence class of \mathcal{E} ;
- Let $X \in C_i$ and set $F = \text{rhs}(X)/\sim$. In case $\text{op}(X) = \wedge$, set $f_i = \wedge F$. Likewise if $\text{op}(X) = \vee$, set $f_i = \vee F$. Set $f_i = C_j$ if $F = \{C_j\}$ and $\text{op}(X) = \perp$;
- Order equations such that $C_i \trianglelefteq C_j$ iff there is some $X \in C_i$ such that for all $X' \in C_j$, $\text{rank}(X) \leq \text{rank}(X')$ and $X \trianglelefteq X'$ in \mathcal{E} .

Properties

Lemma

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. Then $\mathcal{E}' \sim \mathcal{E}_{/\sim}$ implies $\mathcal{E}' \equiv \mathcal{E}_{/\sim}$.

Theorem

The relation \sim is strictly finer than \equiv .

Application

Example

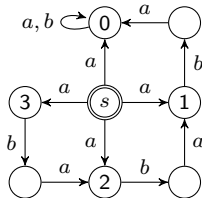
Let N be an arbitrary positive natural number. Consider the process described by the following set of recursive processes (using process-algebraic notation in ACP-style):

$$\left\{ \begin{array}{l} S = \sum \{ a \cdot X(n) \mid n \leq N \}, \\ X(0) = a.X(0) + b.X(0), \quad X(n+1) = b.a.X(n) \end{array} \right\}$$

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 $N = 3$

$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

Equation system for \mathcal{E}_3

Observations:

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

Application (Strong Bisimulation)

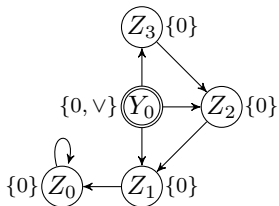
$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

Equation system for \mathcal{E}_3  \mathcal{E}_3 's Dependency Graph

Application (Strong Bisimulation)

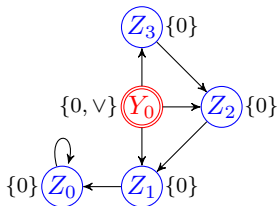
$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

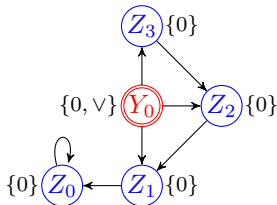
$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

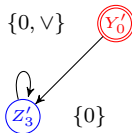
Equation system for \mathcal{E}_3  \mathcal{E}_3 's Dependency Graph

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.

Idempotence identifying Bisimilarity: Definition

Definition

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

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

Equations for X and Y are **strongly bisimilar**, denoted $X \sim Y$, if there is a strong bisimulation relation R such that XRY ;
 \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 $R \subseteq \text{bnd}(\mathcal{E}) \times \text{bnd}(\mathcal{E}')$ is said to be an **idempotence identifying bisimulation** if, whenever XRY , then:

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

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 XRY ;

\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

Definition

Let $\mathcal{E} \in \mathcal{B}$. The **quotient** of \mathcal{E} , denoted \mathcal{E}/\sim_{ii} is defined similar to \mathcal{E}/\sim , except that:

- $\sigma_i C_i = C_j$ in case $\text{rhs}(X_i)/\sim_{ii} = \{C_j\}$.

This quotient avoids introducing awkward equations such as $\sigma_i C_i = \bigwedge \{C_j\}$.

Lemma

Let $\mathcal{E}, \mathcal{E}' \in \mathcal{B}$. Then $\mathcal{E}' \sim_{ii} \mathcal{E}/\sim_{ii}$ implies $\mathcal{E}' \equiv \mathcal{E}/\sim_{ii}$.

Lattice

Theorem

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

The following lemma demonstrates that idempotence identifying bisimilarity and solution equivalence sometimes coincide.

Lemma

Let $\mathcal{E} \in \mathcal{B}$ be of the form $\mathcal{E}_0\mathcal{E}_1\mathcal{E}_2$, with $\mathcal{E}_1 \in \mathcal{B}$. Suppose $X, X' \in \text{bnd}(\mathcal{E}_1)$. Assume $\text{rank}(X) = \text{rank}(X')$ for all $X, X' \in \text{bnd}(\mathcal{E}_1)$. Then $\mathcal{E}_1 / \sim_{ii} = \mathcal{E}_1 / \equiv$.

Idempotence identifying Bisimilarity: Application

Recall our example:

$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

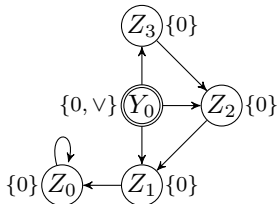
$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

Equation System \mathcal{E}_3



\mathcal{E}_3 's Dependency Graph

Idempotence identifying Bisimilarity: Application

Strong bisimulation:

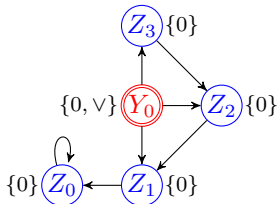
$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

Equation System \mathcal{E}_3  \mathcal{E}_3 's Dependency Graph

Idempotence identifying Bisimilarity: Application

Idempotence identifying bisimulation:

$$(\nu Y_0 = Z_3 \vee Z_2 \vee Z_1)$$

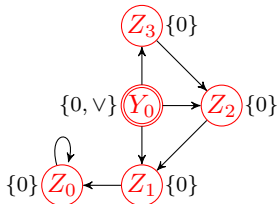
$$(\nu Z_3 = Z_2)$$

$$(\nu Z_2 = Z_1)$$

$$(\nu Z_1 = Z_0)$$

$$(\nu Z_0 = Z_0)$$

Equation System \mathcal{E}_3

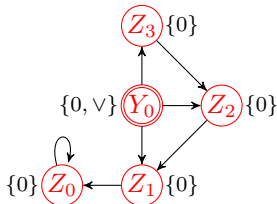


\mathcal{E}_3 's Dependency Graph

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:

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

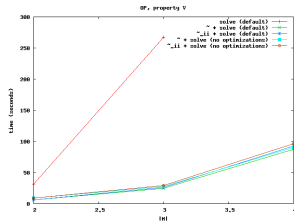
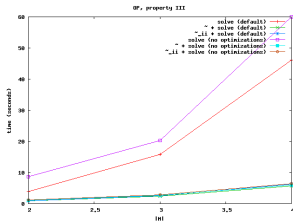
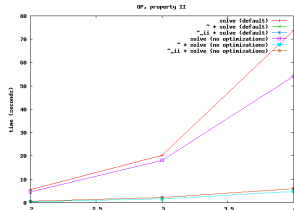
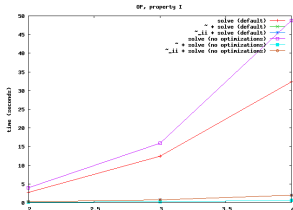
Equation system $\mathcal{E}_{3/\sim_{ii}}$ $\mathcal{E}_{3/\sim_{ii}}$'s dependency graph

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.

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);
- Generalise to Parameterised Boolean Equation Systems.