Analysis of Boolean Equation Systems through Structure Graphs

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

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
1 Introduction

2 Preliminaries

3 Structure Graphs for Boolean Equation Systems
   - Structure Graphs
   - Structure graphs for non-empty closed equation systems

4 Preservation and Reflection of Solution under Bisimilarity
   - true/false-Elimination
   - Normalisation

5 Application

6 Future work
Model Checking

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

- $L$ is a *Labelled Transition System*;
- $f$ is a $\mu$-calculus formula;

Model checking problem = solving Boolean equation systems
A BES is a sequence of fixed point equations
Size BES $\mathcal{E}$ encoding $L \models f$ .......................... $O(|L| \cdot |f|)$
Solving BES = finding the winner in Parity Games
Algorithm solving BES = algorithm for computing winner PG
Computing winner in PG is in UP $\cap$ co-UP .... [Jurdziński’98]
Observations in practice

Minimisation pays ......................... [Keiren & Willemse’09]
- Bisimulation minimisations of underlying dependency graph;
- reductions for closed BESs in standard recursive form;
- Solving $\mathcal{E}$ greatly exceeds minimising $\mathcal{E}$ + solving minimised $\mathcal{E}$;

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

Minimisation efficiently and substantially reduces $n$ and $m$
Contributions of this work

- Graph model for BESs **through** SOS;
- Bisimulation minimisation for **all** closed BESs (not just SRF);
- Identified **impact** of transformation to SRF on minimisation;
1 Introduction

2 Preliminaries

3 Structure Graphs for Boolean Equation Systems
   - Structure Graphs
   - Structure graphs for non-empty closed equation systems

4 Preservation and Reflection of Solution under Bisimilarity
   - true/false-Elimination
   - Normalisation

5 Application

6 Future work
Definition (Boolean Equation System (BES))

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

$$\mathcal{E} ::= \epsilon | (\mu X = f) \mathcal{E} | (\nu X = f) \mathcal{E}$$
$$f, g ::= c | X | f \lor g | f \land g$$

- $\epsilon$ is the empty BES;
- $\mu, \nu$ are least, resp., greatest fixed point signs;
- $X$ is a proposition variable from a set $\mathcal{X}$;
- $c$ is a constant from the set $\{\text{false, true}\}$;

A BES is in SRF if the formulae are given by the following grammar:

$$f ::= X | \lor F | \land F$$

where $F$ is a non-empty set of proposition variables
bound/occurring proposition variables

\[ \text{bnd}(\epsilon) = \emptyset \]
\[ \text{bnd}(\sigma X = f) = \text{bnd}(\mathcal{E}) \cup \{X\} \]
\[ \text{occ}(\epsilon) = \emptyset \]
\[ \text{occ}(\sigma X = f) = \text{occ}(\mathcal{E}) \cup \text{occ}(f) \]
\[ \text{occ}(e) = \emptyset \]
\[ \text{occ}(f \lor g) = \text{occ}(f) \cup \text{occ}(g) \]
\[ \text{occ}(X) = \{X\} \]
\[ \text{occ}(f \land g) = \text{occ}(f) \cup \text{occ}(g) \]

- \(\mathcal{E}\) is closed: \(\text{occ}(\mathcal{E}) \subseteq \text{bnd}(\mathcal{E})\)
- \(X \preceq Y\): the equation for \(X\) precedes \(Y\)'s in \(\mathcal{E}\)
- \(\mathcal{E}\) is well-formed: \(\preceq\) is acyclic;
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 $[[f]] \eta$ maps a proposition formula $f$ to true or false:

- $[[c]] \eta = c$
- $[[f \lor g]] \eta = [[f]] \eta \lor [[g]] \eta$
- $[[X]] \eta = \eta(X)$
- $[[f \land g]] \eta = [[f]] \eta \land [[g]] \eta$
The solution to a BES is defined in context of environment $\eta: \mathcal{X} \rightarrow \mathcal{B}$.

Peculiarities:

- Solution to closed BES independent of environment:

$$\forall X \in \text{bnd}(\mathcal{E}) : \left[ \mathcal{E} \eta(X) = \left[ \mathcal{E} \right] \eta'(X) \right] \text{ for all } \eta, \eta'$$

- Solution is order-sensitive:

$$\left[ (\mu X = Y) \ (\nu Y = X) \right] \neq \left[ (\nu Y = X) \ (\mu X = Y) \right]$$
Example (Mutual exclusion)

On some path a reader can infinitely often start reading:

\[ \nu X. \mu Y. \langle r_s \rangle X \lor \langle r_s \rangle Y \]

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

\[
\begin{align*}
(\nu X_{s_0} &= Y_{s_0}) & (\nu X_{s_1} &= Y_{s_1}) & (\nu X_{s_2} &= Y_{s_2}) & (\nu X_{s_3} &= Y_{s_3}) \\
(\mu Y_{s_0} &= X_{s_1} \lor Y_{s_1}) & (\mu Y_{s_1} &= X_{s_2} \lor Y_{s_0}) & (\mu Y_{s_2} &= Y_{s_1}) & (\mu Y_{s_3} &= Y_{s_0})
\end{align*}
\]
• \text{rank}(X) \text{ indicates in which block of like-signed equations } X \text{ occurs.}

• \text{rank}(X) \text{ is odd iff } X \text{ is defined in a } \mu\text{-equation.}

• \text{rank}(X) \text{ inductively defined on structure of BES.}

\textbf{Example (Rank)}

\begin{align*}
(1) \quad & \mu X = (X \land Y) \lor Z \\
(2) \quad & \nu Y = W \lor (X \land Y) \\
(3) \quad & \mu Z = Z \\
(3) \quad & \mu W = Z \lor (Z \lor W)
\end{align*}
3 Structure Graphs for Boolean Equation Systems
  - Structure Graphs
  - Structure graphs for non-empty closed equation systems

4 Preservation and Reflection of Solution under Bisimilarity
  - true/false-Elimination
  - Normalisation

5 Application

6 Future work
Aim

Identify a graph structure capturing all information needed for solving equation systems.

It should (at least):

- capture **ranks** of bound variables;
- reflect the **dependencies** between bound variables and proposition formulae;
- reflect the **dependencies** between proposition formulae and its subformulae;
- indicate the **top logical operators/ constants** of proposition formulae;
Definition (Structure Graph)

A structure graph is a graph \( G = \langle T, t, \rightarrow, d \rangle \), where:

- \( T \) is a finite set of formulae, and \( t \in T \) is the initial formula;
- \( \rightarrow \subseteq T \times T \) is a dependency relation;
- \( d: T \rightarrow (2^{D_\wedge} \cup 2^{D_\vee} \cup 2^{D_\top} \cup 2^{D_\perp}) \), where, for \( e \in \{\wedge, \vee, \top, \perp\} \), \( D_e = N \cup \{e\} \), is a term decoration mapping;

\( G_i = \langle T_i, t_i, \rightarrow_i, d_i \rangle \); \( R \subseteq T_0 \times T_1 \) is a bisimulation if for all \( u_0 R u_1 \):

- \( d_0(u_0) = d_1(u_1) \);
- for all \( v_0 \in T_0 \), if \( u_0 \rightarrow_0 v_0 \), then \( u_1 \rightarrow_1 v_1 \) for some \( v_1 \in T_1 \) satisfying \( v_0 R v_1 \); (similarly for \( u_1 \rightarrow_1 v_1 \))

\( G_0 \leftrightarrow G_1 \) iff \( t_0 R t_1 \) for some bisimulation \( R \)
Definition

$G = \langle T, t, \rightarrow, d \rangle$ is called BESsy if:

- a node $t$ decorated by $\top$ or $\bot$ has no $\rightarrow$ successor;
- a node is decorated by $\blacktriangle$ or $\blacktriangledown$ or a rank iff it has a $\rightarrow$ successor;
- a node with multiple $\rightarrow$ successors, is decorated with $\blacktriangle$ or $\blacktriangledown$;
- a node with rank 0 or 1 is reachable from $t$,
- the ranks of all reachable nodes form a closed interval

Observation:

- BESsyness is preserved under bisimilarity.
Definition (Transformation BESsy structure graph to BES)

Let $\mathcal{G} = \langle T, t, \rightarrow, d \rangle$ be a BESsy structure graph.

- For each node $u \in T$ with $d(u) \cap \mathcal{N} \neq \emptyset$ introduce an equation:
  \[
  \sigma X_u = \text{rhs}(u) \quad \sigma = \begin{cases} 
  \mu & \text{if } \text{rank}(X_u) \text{ is odd} \\
  \nu & \text{otherwise}
  \end{cases}
  \]
  \[
  \text{rank}(X_u) = \min(d(u) \cap \mathcal{N})
  \]

- The equation system $\mathcal{E}$ is obtained by ordering the equations from left-to-right based on the ranks of the variables;
- The equation system associated to $\mathcal{G}$ is denoted $\mathcal{E}_G$;
- The formula associated to $\mathcal{G}$ is the formula $\text{term}(t)$ in the context of the equation system $\mathcal{E}_G$. 
For BESsy Structure Graphs, \( \text{term} \) and \( \text{rhs} \) are defined as follows:

\[
\text{term}(u) = \begin{cases} 
\bigcap \{ \text{term}(u') \mid u \rightarrow u' \} & \text{if } d(u) = \{ \triangle \}, \\
\biggup \{ \text{term}(u') \mid u \rightarrow u' \} & \text{if } d(u) = \{ \nabla \}, \\
\text{true} & \text{if } \top \in d(u), \\
\text{false} & \text{if } \bot \in d(u), \\
X_u & \text{otherwise,}
\end{cases}
\]

\[
\text{rhs}(u) = \begin{cases} 
\bigcap \{ \text{term}(u') \mid u \rightarrow u' \} & \text{if } \triangle \in d(u), \\
\biggup \{ \text{term}(u') \mid u \rightarrow u' \} & \text{if } \nabla \in d(u), \\
\text{term}(u') & \text{otherwise, } u' \text{ s.t. } u \rightarrow u'.
\end{cases}
\]

\[
\bigcap \{ t \} = t \quad \text{if } t < \min(T), T \neq \emptyset, \text{ then } \bigcap (\{ t \} \cup T) = t \land \left( \bigcap T \right)
\]
Mapping proposition formulae onto a Structure Graph via SOS

- Flatten nesting hierarchy of the same connective:

  \[
  \text{true} \uparrow \quad \text{false} \downarrow \quad (t \land t') \uparrow \quad (t \lor t') \downarrow
  \]

  \[
  t \uparrow \quad t \rightarrow u \quad t \downarrow \quad t \rightarrow u
  \]

  \[
  t \land t' \rightarrow u \quad t \lor t' \rightarrow u
  \]

- Dependencies on variables and constants:

  \[
  \neg t \uparrow \quad \neg t \downarrow
  \]

  \[
  t \land t' \rightarrow t \quad t \lor t' \rightarrow t
  \]
Example \([(X \land (Z \lor (Y \lor X))) \land Z)\]

\[(X \land (Z \lor (Y \lor X))) \land Z \triangleright \rightarrow X\]

\[Z \quad Z \lor (Y \lor X) \triangledown \rightarrow Y\]
Mapping equations onto a Structure Graph via SOS

- Ranking the variables:

\[ X \in \text{bnd}(E) \quad \text{rank}(X) = n \]

\[ X \cap n \]

- The structure of a node representing a variable is derived from the right-hand side of the corresponding equation:

\[ \sigma X = t \in \mathcal{E} \quad \neg t^\Delta \quad \neg t^\nabla \]

\[ X \rightarrow t \]

\[ \sigma X = t \in \mathcal{E} \quad t \rightarrow u \]

\[ X \rightarrow u \]
Example (Equation system and associated structure graph)

\[
\begin{align*}
\mu_X &= (X \land Y) \lor Z \\
\nu_Y &= W \lor (X \land Y) \\
\mu_Z &= Z \\
\mu_W &= Z \lor (Z \lor W)
\end{align*}
\]
Notation:
- $G_{\mathcal{E},t}$ denotes structure graph associated to a formula $t$ in the context of an equation system $\mathcal{E}$
- $G_{\mathcal{E}} = G_{\mathcal{E},X}$, where $X \in bnd(\mathcal{E})$ is the least element w.r.t. $\preceq$

Lemma

Let $\mathcal{E}$ be a non-empty closed equation system. Let $t$, $u$, $v$ be proposition formulae such that $\text{occ}(t) \cup \text{occ}(u) \cup \text{occ}(v) \subseteq bnd(\mathcal{E})$.

$$G_{\mathcal{E},(t \land u) \land v} \Leftrightarrow G_{\mathcal{E},t \land (u \land v)} \quad G_{\mathcal{E},t \land u} \Leftrightarrow G_{\mathcal{E},u \land t}$$

- Idempotency of $\land$ is not captured by bisimilarity
Theorem (Preservation of Solution)

- $\mathcal{E}$ is a non-empty closed BES and $\mathcal{G}_\mathcal{E}$ is its Structure Graph;
- $\mathcal{E}'$ is obtained by transforming $\mathcal{G}_\mathcal{E}$ to BES;

There is a total bijective mapping $h : \text{bnd}(\mathcal{E}) \rightarrow \text{bnd}(\mathcal{E}')$ such that:

$$\forall X \in \text{bnd}(\mathcal{E}) : [\mathcal{E}](X) = [\mathcal{E}'](h(X))$$
1. Introduction

2. Preliminaries

3. Structure Graphs for Boolean Equation Systems
   - Structure Graphs
   - Structure graphs for non-empty closed equation systems

4. Preservation and Reflection of Solution under Bisimilarity
   - true/false-Elimination
   - Normalisation

5. Application

6. Future work
Aim

Bisimilarity preserves and reflects solutions for BESsy structure graphs

Strategy:

\[ G_0 \xrightarrow{\text{reduce}} G'_0 \xrightarrow{\text{normalise}} G''_0 \]
\[ G_1 \xrightarrow{\text{reduce}} G'_1 \xrightarrow{\text{normalise}} G''_1 \]

- reduce removes all constant decorations;
- normalise removes all nestings of ∧ and ∨;
- normalised structure graph is bisimilar to graph for BES in SRF;

Theorem (Keiren & Willemse'09)

Bisimilarity preserves and reflects solution for BES in SRF
Eliminate false and true nodes

- false node maps to (fresh) node $X$ with $\mu X = X$;
- true node maps to (fresh) node $X$ with $\nu X = X$;
- ranks 0, resp. 1 assigned to new nodes;

Operator $\text{reduce}$ is defined by the following deduction rules

\[
\begin{align*}
    & t\Updownarrow \quad t\Downarrow \\
    & \frac{}{\text{reduce}(t)\Updownarrow} \quad \frac{}{\text{reduce}(t)\Downarrow} \\
    & \frac{}{\text{reduce}(t) \to \text{reduce}(\mathit{u})} \\
    & \frac{}{\text{reduce}(t) \uplus \mathit{n}} \\
    & \frac{}{\text{reduce}(t) \to \text{reduce}(t)} \\
    & \frac{}{\text{reduce}(t) \uplus 0} \\
    & \frac{}{\text{reduce}(t) \to \text{reduce}(t)} \\
    & \frac{}{\text{reduce}(t) \uplus 1}
\end{align*}
\]

Theorem (Congruence)

*Bisimilarity is a congruence for $\text{reduce})*
Removing nesting of $\land$ and $\lor$

- Only non-ranked terms occur as subterms in right-hand sides of equations with nested occurrences of $\land$ and $\lor$;
- Rank **non-ranked** nodes based on ranks assigned to successors;

Operator **normalise** is defined by the following deduction rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Antecedent</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t \uparrow$</td>
<td>norm($t$) $\uparrow$</td>
<td>$t$ $\uparrow$</td>
</tr>
<tr>
<td>$t \downarrow$</td>
<td>norm($t$) $\downarrow$</td>
<td>$t$ $\downarrow$</td>
</tr>
<tr>
<td>$t \rightarrow u$</td>
<td>norm($t$) $\rightarrow$ norm($u$)</td>
<td>$t$ $\vdash n$</td>
</tr>
<tr>
<td>$t \not\vdash t \rightarrow u$</td>
<td>norm($u$) $\vdash n$</td>
<td>$\forall v \ t \rightarrow v \Rightarrow$ norm($v$) $\vdash m \land m \leq n$</td>
</tr>
</tbody>
</table>

**Theorem (Congruence)**

*Bisimilarity is a congruence for normalise*
Notation:
- $G_{\text{norm}}(\mathcal{E})$ denotes the structure graph $G_{\mathcal{E},\text{norm}}(X)$ where $X$ is the least variable w.r.t. $\triangleleft$

Lemma (Normalisation preserves and reflects solution)
- $\mathcal{E}$ is a non-empty, closed equation system;
- $\mathcal{E}_{\text{norm}}$ is the equation system obtained by transforming $G_{\text{norm}}(\mathcal{E})$ into an equation system;

There is a total injective mapping $h : \text{bnd}(\mathcal{E}) \rightarrow \text{bnd}(\mathcal{E}_{\text{norm}})$ such that:

$$\forall X \in \text{bnd}(\mathcal{E}) : \llbracket \mathcal{E} \rrbracket(X) = \llbracket \mathcal{E}_{\text{norm}} \rrbracket(h(X)).$$
Example (Normalisation)

\[
\begin{align*}
X \land Y & \uparrow 1 \\
Y & \downarrow 2 \\
X & \downarrow 1 \\
\text{norm}(X \land Y) & \uparrow 2 \\
\text{norm}(Y) & \downarrow 2 \\
\text{norm}(X) & \downarrow 1 \\
\text{norm}(W) & \downarrow 3 \\
Z & \uparrow 3 \\
W & \uparrow 3 \\
\text{norm}(Z) & \downarrow 3 \\
\end{align*}
\]
Theorem

- $\mathcal{E}$ and $\mathcal{E}'$ are non-empty, closed equation systems;
- $f$ w.r.t. $G\mathcal{E}$ and $f'$ w.r.t. $G\mathcal{E}'$ are bisimilar;

\[
[f][\mathcal{E}] = [f'][\mathcal{E}']
\]
1. Introduction
2. Preliminaries
3. Structure Graphs for Boolean Equation Systems
   - Structure Graphs
   - Structure graphs for non-empty closed equation systems
4. Preservation and Reflection of Solution under Bisimilarity
   - true/false-Elimination
   - Normalisation
5. Application
6. Future work
Academic example: LTS modelling an unreliable channel:

- Read a message from the environment \( \text{action } r; \)
- Send a message to the environment \( \text{action } s; \)
- Lose a message \( \text{action } l; \)

Model checking problem

Determine all \( s_i \) that can infinitely often never send a message on all reading/sending paths:

\[
\begin{align*}
{s_i} & \models \nu X. \mu Y. (([r]X \land [s]X \land (\nu Z. \langle \bar{s} \rangle Z)) \lor ([r]Y \land [s]Y))
\end{align*}
\]
The solution to $X_{s_i}$ (in the BES below) answers whether $s_i \models \phi$

$$(\nu X_{s_0} = Y_{s_0}) (\nu X_{s_1} = Y_{s_1}) (\nu X_{s_2} = Y_{s_2})$$
$$(\mu Y_{s_0} = (X_{s_1} \land Z_{s_0}) \lor Y_{s_1}) (\mu Y_{s_1} = (X_{s_0} \land Z_{s_1}) \lor Y_{s_0}) (\mu Y_{s_2} = \text{true})$$
$$(\nu Z_{s_0} = Z_{s_1}) (\nu Z_{s_1} = Z_{s_2}) (\nu Z_{s_2} = Z_{s_1})$$
Bisimulation minimised structure graph:

\[
\begin{align*}
[X_{s0}] &\leftrightarrow 0 \\
[Y_{s0}] &\leftrightarrow ▼ 1 \\
[X_{s1} \land Z_{s0}] &\leftrightarrow ▼\ 1 \\
[X_{s1}] &\leftrightarrow ▼ 1 \\
[Z_{s0}] &\leftrightarrow ▼ 1 \\
\end{align*}
\]
Minimised BES

\[
\begin{align*}
(X_{s_0})/\leftrightarrow 0 & \rightarrow (Y_{s_0})/\leftrightarrow \nabla 1 \rightarrow (X_{s_1} \land Z_{s_0})/\leftrightarrow \Delta \rightarrow (Z_{s_0})/\leftrightarrow 2 \\
(X_{s_2})/\leftrightarrow 0 & \rightarrow (Y_{s_2})/\leftrightarrow 1 \rightarrow \text{true } \top
\end{align*}
\]
Minimised BES:

\[(\nu X[x_{s_0}] / \leftrightarrow = X[y_{s_0}] / \leftrightarrow) \land (\nu X[x_{s_2}] / \leftrightarrow = X[y_{s_2}] / \leftrightarrow) \land (\mu X[y_{s_0}] / \leftrightarrow = (X[x_{s_0}] / \leftrightarrow \land X[z_{s_0}] / \leftrightarrow) \lor X[y_{s_0}] / \leftrightarrow) \lor (\nu X[z_{s_0}] / \leftrightarrow = X[z_{s_0}] / \leftrightarrow) \land (\mu X[y_{s_2}] / \leftrightarrow = \text{true})\]

Observations:

- Answering \( s_i \models \phi \) requires solving 5 equations instead of 9. All equations have true as their solution;
- Original LTS: unreducible using strong bisimulation;
- BES size reduction: roughly 50%;
1 Introduction

2 Preliminaries

3 Structure Graphs for Boolean Equation Systems
   - Structure Graphs
   - Structure graphs for non-empty closed equation systems

4 Preservation and Reflection of Solution under Bisimilarity
   - true/false-Elimination
   - Normalisation

5 Application

6 Future work
Future work:

- Extension to open equation systems and concatenation of such equation systems;
- Extension to other coarser notion of equivalence (stuttering equivalence);