

Algorithms for Model Checking (2IW55)

Lecture 5

Boolean Equation Systems

Background material: Chapter 3 and 6 of

A. Mader, "Verification of Modal Properties using Boolean Equation Systems", Ph.D. thesis, 1997

Tim Willemse

(timw@win.tue.nl)

<http://www.win.tue.nl/~timw>

MF 6.073

Boolean Equation Systems

Model Checking using BESs

Solving BESs

Exercise

- ▶ Boolean Equation Systems are systems of **fixed point equations**.

Given a set Var of **propositional variables**. A Boolean Expression is defined by:

$$f ::= X \mid \text{true} \mid \text{false} \mid f \wedge f \mid f \vee f$$

A **Boolean Equation** is an equation of the form $\mu X = f$ or $\nu X = f$ where $X \in Var$ and f is a Boolean Expression.

A **Boolean Equation System** is a **sequence** of Boolean Equations:

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

Note:

- ▶ Negation is not allowed, in order to ensure monotonicity.
- ▶ The **order** of equations is important. The leftmost sign will be given priority.

- ▶ A variable W that occurs in a Boolean Expression of a BES \mathcal{E} is called **bound**, if there is an equation for W in \mathcal{E} , otherwise W is called **free**.
- ▶ If propositional variables are **bound uniquely** (i.e., at most once), the BES is **well-formed**; we only consider well-formed BESs.
- ▶ If \mathcal{E} contains no free variables, \mathcal{E} is **closed**, otherwise it is **open**.
- ▶ Henceforth, σ represents either μ or ν if we wish to abstract from its actual polarity.

Example

An example of a **closed** BES \mathcal{E} with three propositional variables X , Y and Z :

$$(\mu X = (X \wedge Y) \vee Z) (\nu Y = X \wedge Y) (\mu Z = Z \wedge X)$$

An example of an **open** BES \mathcal{F} with three propositional variables X , Y and Z :

$$(\mu X = Y \vee Z) (\nu Y = X \wedge Y)$$

An example of a BES that is not well-formed:

$$(\mu X = X) (\nu X = X)$$

- ▶ Let Val be the set of all functions $\eta : Var \rightarrow \{\text{false}, \text{true}\}$
- ▶ The **solution** of a BES is a valuation: $\eta : Val$
- ▶ Let $[f](\eta)$ denote the **value** of boolean expression f under valuation η .
- ▶ For the solution η of a BES \mathcal{E} , we wish $\eta(X) = [f](\eta)$ for all equations $\sigma X = f$ in \mathcal{E} .
- ▶ Also, we want the smallest (for μ) or greatest (for ν) solution, where leftmost fixed point signs take priority over fixed point signs that follow.

Given a BES \mathcal{E} , we define $\llbracket \mathcal{E} \rrbracket : Val \rightarrow Val$ by recursion on \mathcal{E} .

$$\left\{ \begin{array}{l} \llbracket \mathcal{E} \rrbracket(\eta) \quad \quad \quad := \eta \\ \llbracket (\mu X = f) \mathcal{E} \rrbracket(\eta) \quad := \llbracket \mathcal{E} \rrbracket(\eta[X := [f](\eta_\mu)]) \text{ where } \eta_\mu := \llbracket \mathcal{E} \rrbracket(\eta[X := \text{false}]) \\ \llbracket (\nu X = f) \mathcal{E} \rrbracket(\eta) \quad := \llbracket \mathcal{E} \rrbracket(\eta[X := [f](\eta_\nu)]) \text{ where } \eta_\nu := \llbracket \mathcal{E} \rrbracket(\eta[X := \text{true}]) \end{array} \right.$$

Note: for closed BESs we have $\llbracket \mathcal{E} \rrbracket(\eta)(X) = \llbracket \mathcal{E} \rrbracket(\eta')(X)$ for all η, η' and all bound X

Boolean Equation Systems

Model Checking using BESs

Solving BESs

Exercise

Transformation of the μ -calculus model checking problem to BES

- ▶ Given is the following model checking problem: $M, t \models \sigma X. f$
 - a closed μ -calculus formula $\sigma X. f$ in **Positive Normal Form** and,
 - a Mixed Kripke Structure $M = \langle S, s_0, Act, R, L \rangle$.
 - $t \in S$ is a state
- ▶ We define a BES \mathcal{E} with the following property:

$$([\mathcal{E}](\eta))(X_t) = \text{true iff } M, t \models \sigma X. f$$

i.e. formula $\sigma X. f$ holds in state t if and only if the solution for X_t yields true.

- ▶ This BES is defined as follows:
 - For each subformula $\sigma' Y. g$, we add the following equation for each state $t \in S$:
$$\sigma' Y_t = \text{RHS}(t, g)$$
 - **Important:** The order of the equations respects the subterm ordering in the original formula $\sigma X. f$.
 - **Intuitively:** We wish $\text{RHS}(t, g)$ iff $t \models g$

The **Right-Hand Side** of an equation is defined inductively on the structure of the μ -calculus formula:

$$RHS(t, \text{true}) = \text{true}$$

$$RHS(t, \text{false}) = \text{false}$$

$$RHS(t, p) = \begin{cases} \text{true} & \text{if } p \in L(t) \\ \text{false} & \text{otherwise} \end{cases}$$

$$RHS(t, X) = X_t$$

$$RHS(t, f \wedge g) = RHS(t, f) \wedge RHS(t, g)$$

$$RHS(t, f \vee g) = RHS(t, f) \vee RHS(t, g)$$

$$RHS(t, [a]f) = \bigwedge_{u \in S} \{RHS(u, f) \mid t \xrightarrow{a} u\}$$

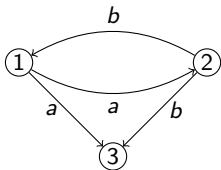
$$RHS(t, \langle a \rangle f) = \bigvee_{u \in S} \{RHS(u, f) \mid t \xrightarrow{a} u\}$$

$$RHS(t, \mu X. f) = X_t$$

$$RHS(t, \nu X. f) = X_t$$

conventions: $\bigwedge_{t \in S} \emptyset = \text{true}$ and $\bigvee_{t \in S} \emptyset = \text{false}$

Example

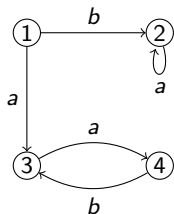


- ▶ $RHS(1, [a]X) = RHS(2, X) \wedge RHS(3, X) = X_2 \wedge X_3.$
- ▶ $RHS(2, \langle b \rangle Y) = RHS(1, Y) \vee RHS(3, Y) = Y_1 \vee Y_3.$
- ▶ $RHS(3, \langle b \rangle Y) = \text{false}$ (empty disjunction!)
- ▶
$$\begin{aligned} & RHS(1, [a]\langle b \rangle \mu Z. Z) \\ &= RHS(2, \langle b \rangle \mu Z. Z) \wedge RHS(3, \langle b \rangle \mu Z. Z) \wedge \\ &= (RHS(1, \mu Z. Z) \vee RHS(3, \mu Z. Z)) \wedge \text{false} \\ &= (Z_1 \vee Z_3) \wedge \text{false} \end{aligned}$$
- ▶ Translation of $\mu X. \langle b \rangle \text{true} \vee \langle a \rangle X$ to BES:

$$(\mu X_1 = X_3 \vee X_2) (\mu X_2 = \text{true}) (\mu X_3 = \text{false})$$

Example

μ -calculus formula: $\nu X.([a]X \wedge \nu Y.\mu Z.(\langle b \rangle Y \vee \langle a \rangle Z))$
 Translates to the following BES:



$$\begin{aligned}
 \nu X_1 &= X_3 \wedge Y_1 \\
 \nu X_2 &= X_2 \wedge Y_2 \\
 \nu X_3 &= X_4 \wedge Y_3 \\
 \nu X_4 &= \text{true} \wedge Y_4 \\
 \nu Y_1 &= Z_1 \\
 \nu Y_2 &= Z_2 \\
 \nu Y_3 &= Z_3 \\
 \nu Y_4 &= Z_4 \\
 \mu Z_1 &= Y_2 \vee Z_3 \\
 \mu Z_2 &= \text{false} \vee Z_2 \\
 \mu Z_3 &= \text{false} \vee Z_4 \\
 \mu Z_4 &= Y_3 \vee \text{false}
 \end{aligned}$$

Boolean Equation Systems

Model Checking using BESs

Solving BESs

Exercise

- ▶ We reduced the model checking problem $M, s \models f$ to the solution of a BES with $\mathcal{O}(|M| \times |f|)$ equations.
- ▶ We now want a **fast procedure** to solve such BESs.
- ▶ An extremely tedious way to solve a BES is to unfold its semantics.
- ▶ A very appealing solution is to solve it by Gauß Elimination.

Gauß Elimination uses the following 4 basic operations to solve a BES:

- ▶ **local solution**: eliminate X in its defining equation:

$$\begin{array}{l} \mathcal{E}_0 (\mu X = f) \mathcal{E}_1 \quad \text{becomes} \quad \mathcal{E}_0 (\mu X = f[X := \text{false}]) \mathcal{E}_1 \\ \mathcal{E}_0 (\nu X = f) \mathcal{E}_1 \quad \text{becomes} \quad \mathcal{E}_0 (\nu X = f[X := \text{true}]) \mathcal{E}_1 \end{array}$$

- ▶ Substitute **definitions to the left**:

$$\begin{array}{l} \mathcal{E}_0 (\sigma_1 X = X \vee Y) \mathcal{E}_1 (\sigma_2 Y = Y \wedge X) \mathcal{E}_2 \\ \text{becomes:} \quad \mathcal{E}_0 (\sigma_1 X = X \vee (Y \wedge X)) \mathcal{E}_1 (\sigma_2 Y = Y \wedge X) \mathcal{E}_2 \end{array}$$

- ▶ Substitute **closed equations to the right**:

$$\begin{array}{l} \mathcal{E}_0 (\sigma_1 X = \text{true}) \mathcal{E}_1 (\sigma_2 Y = Y \wedge X) \mathcal{E}_2 \\ \text{becomes:} \quad \mathcal{E}_0 (\sigma_1 X = \text{true}) \mathcal{E}_1 (\sigma_2 Y = Y \wedge \text{true}) \mathcal{E}_2 \end{array}$$

- ▶ **Boolean simplification**: At least the following:

$$b \wedge \text{true} \rightarrow b \quad b \vee \text{true} \rightarrow \text{true} \quad b \wedge \text{false} \rightarrow \text{false} \quad b \vee \text{false} \rightarrow b$$

Example

local \rightarrow

$$(\mu X = X \vee Y) (\nu Y = X \vee (Y \wedge Z)) (\mu Z = Y \wedge Z)$$

simplifications \rightarrow

$$(\mu X = \text{false} \vee Y) (\nu Y = X \vee (\text{true} \wedge Z)) (\mu Z = Y \wedge \text{false})$$

substitution backwards \rightarrow

$$(\mu X = Y) (\nu Y = X \vee Z) (\mu Z = \text{false})$$

simplifications \rightarrow

$$(\mu X = Y) (\nu Y = X \vee \text{false}) (\mu Z = \text{false})$$

substitution backwards \rightarrow

$$(\mu X = Y) (\nu Y = X) (\mu Z = \text{false})$$

local \rightarrow

$$(\mu X = X) (\nu Y = X) (\mu Z = \text{false})$$

substitution to the right \rightarrow

$$(\mu X = \text{false}) (\nu Y = X) (\mu Z = \text{false})$$

$$(\mu X = \text{false}) (\nu Y = \text{false}) (\mu Z = \text{false})$$

Gauß Elimination is a decision procedure for computing the solution to a BES.

Input: a BES $(\sigma_1 X_1 = f_1) \dots (\sigma_n X_n = f_n)$. Returns: the solution for X_1 .

```

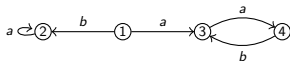
for  $i = n$  downto 1 do
  if  $\sigma_i = \mu$  then  $f_i := f_i[X_i := \text{false}]$ 
  else  $f_i := f_i[X_i := \text{true}]$ 
  end if
  for  $j = i - 1$  downto 1 do  $f_j := f_j[X_i := f_i]$ 
  end for
end for

```

Note:

- ▶ **Invariants** of the outer loop:
 - f_i contains only variables X_j with $j \leq i$.
 - for all $i < j \leq n$, X_j does not occur in f_j .
- ▶ Upon termination ($i = 0$), $\sigma_1 X_1 = f_1$ is **closed** and evaluates to true or false.
- ▶ One could **substitute** the solution for X_1 **to the right** and repeat the procedure to solve X_2 , *etcetera*.

Example



Encoding the μ -calculus formula: $\nu X.([a]X \wedge \nu Y. \mu Z. (\langle b \rangle Y \vee \langle a \rangle Z))$ leads to the below BES; solving using Gauß Elimination (each column is one iteration of the algorithm):

$\nu X_1 = X_3 \wedge Y_1$	$X_3 \wedge Y_1$	$X_3 \wedge Y_1$	$X_3 \wedge Y_1$	$X_3 \wedge Y_1$	$X_3 \wedge Y_1$	$X_3 \wedge Y_1$...	true
$\nu X_2 = X_2 \wedge Y_2$	$X_2 \wedge Y_2$	$X_2 \wedge Y_2$	$X_2 \wedge Y_2$	$X_2 \wedge Y_2$	$X_2 \wedge Y_2$	$X_2 \wedge Y_2$...	false
$\nu X_3 = X_4 \wedge Y_3$	$X_4 \wedge Y_3$	$X_4 \wedge Y_3$	$X_4 \wedge Y_3$	$X_4 \wedge Y_3$	$X_4 \wedge Y_3$	$X_4 \wedge Y_3$...	true
$\nu X_4 = Y_4$	Y_4	Y_4	Y_4	Y_4	Y_4	Y_3	...	true
$\nu Y_1 = Z_1$	Z_1	Z_1	Z_1	Z_1	$Y_2 \vee Y_3$	$Y_2 \vee Y_3$...	true
$\nu Y_2 = Z_2$	Z_2	Z_2	false	false	false	false	...	false
$\nu Y_3 = Z_3$	Z_3	Y_3	Y_3	Y_3	Y_3	Y_3	...	true
$\nu Y_4 = Z_4$	Y_3	Y_3	Y_3	Y_3	Y_3	Y_3^*	...	true
$\mu Z_1 = Y_2 \vee Z_3$	$Y_2 \vee Z_3$	$Y_2 \vee Y_3$	$Y_2 \vee Y_3$	$Y_2 \vee Y_3$	$Y_2 \vee Y_3^*$	$Y_2 \vee Y_3^*$...	true
$\mu Z_2 = Z_2$	Z_2	Z_2	false*	false*	false*	false*	...	false
$\mu Z_3 = Z_4$	Y_3	Y_3^*	Y_3^*	Y_3^*	Y_3^*	Y_3^*	...	true
$\mu Z_4 = Y_3$	Y_3^*	Y_3^*	Y_3^*	Y_3^*	Y_3^*	Y_3^*	...	true

Complexity of Gauß Elimination.

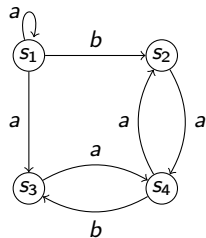
- ▶ Note that in $\mathcal{O}(n^2)$ substitutions, we obtain the final answer for X_1 .
- ▶ However, f_1 can have $\mathcal{O}(2^n)$ different copies of e_n as subterms, so intermediate expressions could become exponentially big.
- ▶ Practical efficiency increases a lot if one keeps all intermediate terms **simplified** all the time.
- ▶ Gauß Elimination can be sped up if a **forward dependency analysis** is conducted (so-called **local model checking**).
- ▶ Precise efficiency **depends heavily** on the set of simplification rules.
- ▶ Precise complexity of solving Boolean Equation Systems is still **unknown**.
- ▶ Complexity of Gauß Elimination is **independent** of the alternation depth (see Proposition 6.4 [Mader]).

Boolean Equation Systems

Model Checking using BESs

Solving BESs

Exercise



Consider the following μ -Calculus formula f :

$$\nu X. ([a]X \wedge \nu Y. \mu Z. (\langle b \rangle Y \vee \langle a \rangle Z))$$

- ▶ Use the Emerson-Lei algorithm for computing whether $M, s_1 \models f$.
- ▶ Translate the model checking question $M \models f$ to a BES; indicate how $M, s \models \phi$ corresponds to the variables in the BES.
- ▶ Solve the BES by Gauß Elimination.