

Algorithms for Model Checking (2IMF35)

Lecture 11

Parameterised Boolean Equation Systems (3)

Background material:

- *Verification of Reactive Systems via Instantiation of Parameterised Boolean Equation Systems*, B. Ploeger, J.W. Wesselink and T.A.C. Willemse (*I&C 2010/2011*)
- *Static Analysis Techniques for Parameterised Boolean Equation Systems*, S. Orzan, J.W. Wesselink and T.A.C. Willemse (*TACAS 2009*)

Tim Willemse

(timw@win.tue.nl)

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

MF 6.073

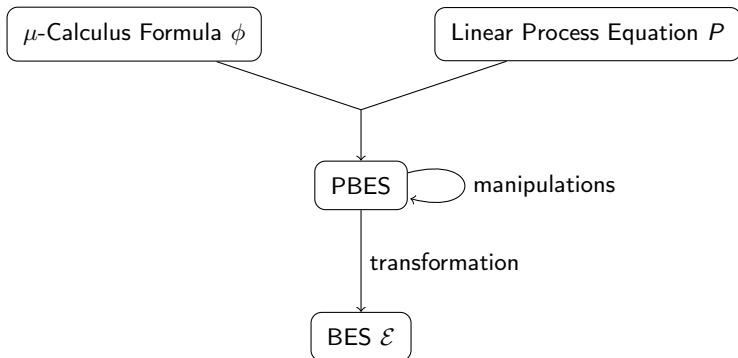
Parameterised Boolean Equation Systems

Instantiation

Manipulations

Exercise

Verification Methodology:



Solving \mathcal{E} answers $P \models \phi$

Problem Description

1. Given a process $P(e)$ described by an LPE P over Act
2. Given a first-order modal μ -calculus formula $\sigma X.\phi$
3. Given environments η, ε
4. Check whether $P(e) \models \sigma X.\phi$ holds, where:

$$P(e) \models \sigma X.\phi \text{ iff } e \in \llbracket \sigma X.\phi \rrbracket_{\eta\varepsilon}$$

5. Conversion to PBES:

$$P(e) \models \sigma X.\phi \text{ iff } e \in \llbracket E(\sigma X.\phi) \rrbracket_{\eta\varepsilon(\tilde{X})} \text{ (or, more informally: } \tilde{X}(e) = \text{true)}$$

Parameterised Boolean Equation Systems

Instantiation

Manipulations

Exercise

How to solve PBESs

$$X_i(e) \stackrel{?}{=} \text{true in } \mathcal{E} := (\sigma_1 X_1(d_1 : D_1) = \phi_1) \cdots (\sigma_n X_n(d_n : D_n) = \phi_n)$$

Known techniques for solving/simplifying \mathcal{E} :

- ▶ Gauß Elimination on PBES + symbolic approximation of equations
- ▶ **Instantiation to BES and subsequently solve the BES**
- ▶ Using patterns
- ▶ Using under/over approximation
- ▶ Invariants

Definition (Logical Equivalence)

Let ϕ, ψ be two predicates. Then ψ is logically equivalent to ϕ , denoted $\phi \leftrightarrow \psi$ iff

$$\forall \varepsilon, \eta : \llbracket \phi \rrbracket \eta \varepsilon = \llbracket \psi \rrbracket \eta \varepsilon$$

- ▶ If $\phi \leftrightarrow \psi$, then equation $\nu X(d : D) = \phi$ has the same solution as $\nu X(d : D) = \psi$ (likewise for μ)
- ▶ Useful simplifications:
 - $\text{false} \wedge \phi \leftrightarrow \text{false}$
 - $\text{true} \vee \phi \leftrightarrow \text{true}$
 - if $d \notin \text{FV}(\phi)$, then $(\exists d : D. \phi) \leftrightarrow (\forall d : D. \phi) \leftrightarrow \phi$
 - One-point rule: $(\exists d : D. d = e \wedge \phi(d)) \leftrightarrow \phi(e)$
 - One-point rule: $(\forall d : D. d = e \Rightarrow \phi(d)) \leftrightarrow \phi(e)$
- ▶ Apply logical simplifications **before** applying PBES manipulations/solving techniques.

Instantiation to BES:

$$X_i(e) \stackrel{?}{=} \text{true in } \mathcal{E} := (\sigma_1 X_1(d_1 : D_1) = \phi_1) \cdots (\sigma_n X_n(d_n : D_n) = \phi_n)$$

- ▶ Let X_i^e be a fresh propositional variable representing instance $X_i(e)$.
- ▶ The procedure below creates a BES from \mathcal{E} s.t. $X_i(e) = \text{true}$ iff $X_i^e = \text{true}$
 1. For each $X_j(e_j)$ occurring in $\text{eval}(\phi_i[d_i := e])$ create a fresh variable $X_j^{e_j}$
 2. Create an equation $\sigma_i X_i^e = \tilde{\phi}_i$, where:
 - $\overline{\phi}_i = \text{eval}(\phi_i[d_i := e])$,
 - $\tilde{\phi}_i$ is $\overline{\phi}_i$ in which every $X_j(e_j)$ is replaced by $X_j^{e_j}$
 3. Repeat step 1 and 2 for every $X_j^{e_j}$ introduced in step 1, for which there is no equation
 4. Order all equations $\sigma_i X_i^e = \dots$ according to the ordering of \mathcal{E} (ordering **within** a block may be arbitrary)

Example

PBES: $(\nu X(n : \text{Nat}) = n \leq 2 \wedge Y(n)) (\mu Y(n : \text{Nat}) = \text{odd}(n) \vee X(n + 1))$

Instantiation starting at e.g. $X(0)$ introduce X^0

1. $Y(0)$ occurs in $\text{eval}((n \leq 2 \wedge Y(n))[n := 0])$ introduce Y^0
2. Introduce $\nu X^0 = \text{eval}(0 \leq 2 \wedge Y^0)$ $\nu X^0 = Y^0$
3. $X(1)$ occurs in $\text{eval}((\text{odd}(n) \vee X(n + 1))[n := 0])$ introduce X^1
4. Introduce $\mu Y^0 = \text{eval}(\text{odd}(0) \vee X^1)$ $\mu Y^0 = X^1$
5. $Y(1)$ occurs in $\text{eval}((n \leq 2 \wedge Y(n))[n := 1])$ introduce Y^1
6. Introduce $\nu X^1 = \text{eval}(1 \leq 2 \wedge Y^1)$ $\nu X^1 = Y^1$
7. no variable occurs in $\text{eval}((\text{odd}(n) \vee X(n + 1))[n := 1])$ end
8. Introduce $\mu Y^1 = \text{eval}(\text{odd}(1) \vee X^2)$ $\mu Y^1 = \text{true}$
9. Order equations: first X^i , then Y^j
10. Resulting BES: $(\nu X^0 = Y^0) (\nu X^1 = Y^1) (\mu Y_0 = X^1) (\mu Y_1 = \text{true})$

Parameterised Boolean Equation Systems

Instantiation

Manipulations

Exercise

Definition (Simple Formula)

A **simple formula** is a formula not containing predicate variables

Observations:

1. Consider the equation $\nu X(n : \text{Nat}) = \text{true} \wedge X(n + 1)$
 - X has solution Nat (check!)
 - Consider formal parameter n :
 - It does not affect the value of the **simple subformula** true
 - It appears to be **redundant** for the solution to X
2. Consider the equation $\nu X(n : \text{Nat}, m : \text{Nat}) = n \leq 5 \wedge X(n + m, m)$
 - X has solution $\{(n, 0) \in \text{Nat} \times \text{Nat} \mid n \leq 5\}$ (check!)
 - Consider formal parameter m :
 - It does not affect the value of the **simple formula** $n \leq 5$
 - Via a single recursion through X , it **does** affect the value of $n \leq 5$
 - It appears to become **significant** for the solution to X

- ▶ Identify all **obvious significant** formal parameters sig
- ▶ Identify the **dependencies** dep

$$\text{sig}(b) = \text{FV}(b)$$

$$\text{sig}(X(e)) = \emptyset$$

$$\text{sig}(\phi \wedge \psi) = \text{sig}(\phi) \cup \text{sig}(\psi)$$

$$\text{sig}(\phi \vee \psi) = \text{sig}(\phi) \cup \text{sig}(\psi)$$

$$\text{sig}(\forall d:D. \phi) = \text{sig}(\phi) \setminus \{d\}$$

$$\text{sig}(\exists d:D. \phi) = \text{sig}(\phi) \setminus \{d\}$$

$$\text{dep}(b) = \emptyset$$

$$\text{dep}(X(e)) = \{X(e)\}$$

$$\text{dep}(\phi \wedge \psi) = \text{dep}(\phi) \cup \text{dep}(\psi)$$

$$\text{dep}(\phi \vee \psi) = \text{dep}(\phi) \cup \text{dep}(\psi)$$

$$\text{dep}(\forall d:D. \phi) = \text{dep}(\phi)$$

$$\text{dep}(\exists d:D. \phi) = \text{dep}(\phi)$$

Examples:

- ▶ $\text{sig}(\text{true} \wedge X(n+1)) = \emptyset$, $\text{sig}(n \leq 5 \wedge X(n+m, m)) = \{n\}$
- ▶ $\text{dep}(\text{true} \wedge X(n+1)) = \{X(n+1)\}$, $\text{dep}(n \leq 5 \wedge X(n+m, m)) = \{X(n+m, m)\}$

Assume the following PBES:

$$\mathcal{E} := (\sigma_1 X_1(d_1 : D_1) = \phi_1) \cdots (\sigma_n X_n(d_n : D_n) = \phi_n)$$

- ▶ $\text{arity}(X_i)$: the length of vector d_i
- ▶ $d_i[j]$ denotes the j -th element of vector d_i
- ▶ Construct a **marked influence graph** $G(\mathcal{E}) = \langle V, \longrightarrow, M \rangle$:
 - ▶ $V = \{(X_i, j) \mid 1 \leq j \leq \text{arity}(X_i)\}$ is the set of **vertices**
 - ▶ $(X_i, k) \longrightarrow (X_j, l)$ iff for some expression e : $X_j(e) \in \text{dep}(\phi_i)$ and $d_i[k] \in \text{FV}(e[l])$
 - ▶ $M = \{(X_i, j) \mid 1 \leq i \leq n \text{ and } d_i[j] \in \text{sig}(\phi_i)\}$ is the **marking**

Definition (Positively redundant parameters)

Given a Marked Influence Graph $G(\mathcal{E}) = \langle V, \longrightarrow, M \rangle$.

The set of **positively redundant parameters** of \mathcal{E} is:

$$\mathcal{R} = \{d_i[j] \mid \neg(\exists (X_k, l) \in M : (X_i, j) \longrightarrow^* (X_k, l))\}$$

- ▶ Computing the set \mathcal{R} requires $\mathcal{O}(|\longrightarrow|)$ steps at most
- ▶ \mathcal{R} can be computed using a standard least fixed point computation, a depth-first search or a breadth-first search.

Given closed equation system \mathcal{E} with no unbound data variables

Procedure for eliminating redundant parameters in \mathcal{E}

1. Step 1 (compute redundant parameters)
 - 1.1 Construct Marked Influence Graph of \mathcal{E}
 - 1.2 Compute the set \mathcal{R} of positive redundant parameters of \mathcal{E}
2. Step 2 (remove redundant parameters): for every equation $\sigma_i X_i(d_i; D_i) = \phi_i$ in \mathcal{E} :
 - 2.1 remove parameter $d_i[j]$ from $X_i(d_i; D_i)$ iff $d_i[j] \in \mathcal{R}$
 - 2.2 remove expression $e[j]$ from an occurrence $X_k(e)$ in ϕ_i iff $d_k[j] \in \mathcal{R}$

Theorem (Redundancy)

*The modified equation system \mathcal{E} has the “same” solution as \mathcal{E} , i.e., the solution of a variable X **does not depend** on the parameters that have been identified as positively redundant.*

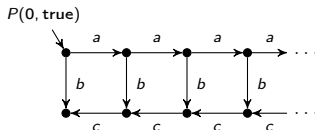
Example

- ▶ $\nu X(b:Bool, n:Nat) = b \wedge X(b, n + 1)$ has solution $f = \{(c, v) \in Bool \times Nat \mid c = \text{true}\}$
- ▶ $\nu X(b:Bool) = b \wedge X(b)$ has solution $g = \{c \in Bool \mid c = \text{true}\}$
- ▶ For all $c \in Bool, v \in Nat$, $(c, v) \in f$ iff $c \in g$.

Example

Consider the following process:

$$\begin{aligned}
 & P(n: \text{Nat}, d: \text{Bool}) \\
 = & d \longrightarrow a \cdot P(n+1, d) \\
 + & d \longrightarrow b \cdot P(n, \neg d) \\
 + & \neg d \wedge n > 0 \longrightarrow c \cdot P(n-1, d)
 \end{aligned}$$



Along every a path, always a b action is attainable:

$$\nu V. ([a]V \wedge \mu W. (\langle a \rangle W \vee \langle b \rangle \text{true}))$$

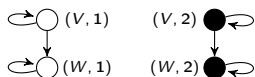
$$\begin{aligned}
 \text{PBES: } \left\{ \begin{aligned}
 \nu V(n: \text{Nat}, d: \text{Bool}) &= (d \implies V(n+1, d)) \wedge W(n, d) \\
 \mu W(n: \text{Nat}, d: \text{Bool}) &= d \vee (d \wedge W(n+1, d))
 \end{aligned} \right.
 \end{aligned}$$

Instantiation of the PBES does not terminate.

Example (Cont'd)

$$\text{PBES: } \begin{cases} (\nu V(n:\text{Nat}, d:\text{Bool}) = (d \implies V(n+1, d)) \wedge W(n, d)) \\ (\mu W(n:\text{Nat}, d:\text{Bool}) = d \vee (d \wedge W(n+1, d))) \end{cases}$$

- ▶ $\text{dep}((d \implies V(n+1, d)) \wedge W(n, d)) = \{V(n+1, d), W(n, d)\}$
- ▶ $\text{dep}(d \vee (d \wedge W(n+1, d))) = \{W(n+1, d)\}$
- ▶ Marked Influence Graph ($\text{arity}(V) = \text{arity}(W) = 2$):



Marked states are black;
non-marked white

- ▶ $\mathcal{R} = \{(V, 1), W(1)\}$, i.e., parameter n is positively redundant for V and W .
- ▶ Reduced PBES:
$$\begin{cases} (\nu V(d:\text{Bool}) = (d \implies V(d)) \wedge W(d)) \\ (\mu W(d:\text{Bool}) = d \vee (d \wedge W(d))) \end{cases}$$
- ▶ Instantiation of the above PBES **terminates**

Parameterised Boolean Equation Systems

Instantiation

Manipulations

Exercise

Consider the lossy channel system described by the following LPE:

$$\begin{aligned}
 C(b : Bool, m : M) &= \sum_{k:M} b \longrightarrow r(k) \cdot C(\text{false}, k) \\
 &+ \neg b \longrightarrow s(m) \cdot C(\text{true}, m) \\
 &+ \neg b \longrightarrow l \cdot C(\text{true}, m)
 \end{aligned}$$

Action r stands for reading, s stands for sending and l stands for losing a message.

- $\nu X.([\text{true}]X \wedge (\mu Y.[l]Y \wedge \forall m:M.[r(m)]Y \wedge \langle \text{true} \rangle \text{true}))$
- $\nu X.\mu Y.\nu Z.(\forall m:M.[s(m)]X) \wedge ((\forall m:M.[s(m)]\text{false}) \vee ([l]Y \wedge \forall m:M.[r(m)]Y)) \wedge [l]Z \wedge \forall m:M.[r(m)]Z$

Questions:

- ▶ Translate both formulae to PBESs given process $C(\text{true}, m_0)$
- ▶ Use instantiation to compute BESs when $M = Bool$, and solve the BES ($m_0 = \text{true}$)
- ▶ Can you remove redundant parameters? If so, remove these redundant parameters and try instantiation to compute a BES when $M = Nat$ ($m_0 = 0$)