Abstraction in Fixpoint Logics

Tim Willemse (TU/e)

Joint work with Maciek Gazda, Sjoerd Cranen, Wieger Wesselink

May 9, 2014

Outline

Fixpoint Logics for Verification

Boolean Equation Systems

Abstraction

Conclusions and Outlook
Fixpoint Logics for Verification

Specs \rightarrow(Encoding) \rightarrow(fixpoint logic)

Semantics \rightarrow(L; formula f)

L \models \phi \rightarrow(BES, PBES, LFP, ...)

---

Fixpoint Logics for Verification

Specs \rightarrow(Encoding) \rightarrow(fixpoint logic)

Semantics \rightarrow(L_1 \sim L_2)

L_1 \models \phi \rightarrow(BES, PBES, LFP, ...)

---
Example (All Work...)

Always $w(ork)$...

$X \overset{\nu}{=} w \land \Box X$
Example (All Work...)

Compute where $X$ holds by approximation:

- $X^0 = \{s_0, s_1, s_2, s_3\}$
- $X^1 = \{s_0, s_2, s_3\}$
- $X^2 = \{s_0, s_3\}$
- $X^3 = \emptyset$

\[
x = \nu \: w \land \Box X
\]

\[
Y = \mu \: (w \land \Box X) \lor (p \land \Box Y)
\]
Fixpoint Logics for Verification

Example (All Work...)

\[
\begin{align*}
X & \equiv Y \\
Y & \equiv (w \land \Box X) \lor (p \land \Box Y)
\end{align*}
\]

Convert to Boolean Equation System:

\[
\begin{align*}
X_{s0} & \equiv Y_{s0} \\
X_{s1} & \equiv Y_{s1} \\
X_{s2} & \equiv Y_{s2} \\
X_{s3} & \equiv Y_{s3} \\
Y_{s0} & \equiv X_{s2} \land X_{s3} \\
Y_{s1} & \equiv Y_{s0} \\
Y_{s2} & \equiv X_{s1} \\
Y_{s3} & \equiv X_{s2}
\end{align*}
\]

\(X_{si}\) is true iff \(s_i \in X\); same for \(Y\)

Outline

- Fixpoint Logics for Verification
- Boolean Equation Systems
- Abstraction
- Conclusions and Outlook
A Boolean Equation is an equation of the form

\[ X \equiv \mu f_X \]

fixpoint equality; can also be \( \nu \)

propositional variable
A Boolean Equation is an equation of the form

\[ X \mu f_X \]

propositional formula; propositional variables occur only positively.

Semantics: the least (resp. largest) Boolean satisfying the equation.
A **Boolean Equation** is an equation of the form

\[ X \overset{\mu}{=} f_X \]

Semantics: the least (resp. largest) Boolean satisfying the equation.

**Example**

- \( X \overset{\mu}{=} \text{true} \): solution to \( X \) is *true*
- \( X \overset{\mu}{=} X \): solution to \( X \) is *false*
- \((X \overset{\mu}{=} Y)\): solution to \( X \) is determined by \( Y \).

A **Boolean Equation System** is a **sequence** of the form

\[
( X_1 \overset{\sigma_1}{=} f_1 ) \cdot \cdot \cdot ( X_n \overset{\sigma_n}{=} f_n )
\]

Semantics assigns a solution to each predicate variable

\[
[\emptyset] \theta = \theta \quad [(X \overset{\sigma}{=} f)f] \theta \ = \ [E] \theta [X := F_{\sigma}]
\]

where \( \theta \) is a **propositional environment** and:

\( F_{\mu}/F_{\nu} \) is the least/greatest \( F \) satisfying: \( F = [f]([E] \theta [X := F]) \)
Outline

Fixpoint Logics for Verification

Boolean Equation Systems

Abstraction

Conclusions and Outlook

Abstraction

Efficient algorithms for solving BESs:
  • Gauß Elimination or directly via semantics
  • Parity game solvers: Zielonka’s algorithm, Small Progress Measures

Succinct representation of large/infinite BESs:
  • PBESs: first-order logic + fixpoints

How to solve large/infinite BESs?
Abstraction

Example (All Work...)

\[ \begin{align*} 
X & \equiv Y \\
Y & \equiv (w \land \Box X) \lor (p \land \Box Y) 
\end{align*} \]

Infinite Boolean Equation System:

\[ \begin{align*} 
X_{s0} & \equiv Y_{s0} \\
X_{s1} & \equiv Y_{s1} \\
X_{s2} & \equiv Y_{s2} \\
X_{s3} & \equiv Y_{s3} \\
\vdots & \quad \vdots 
\end{align*} \]

Abstraction

How to deal with infinite BESs:

- In process theory...................... simulation relations
- In propositional logic................. logical consequence

Definition (Consistent Consequence)

Essentially, add the following rule to a proof system for logical consequence:

\[ \Gamma \cup \{X \Rightarrow Y\} \vdash f_X \Rightarrow f_Y \quad \text{block}(X) = \text{block}(Y) \]

\[ \Gamma \vdash X \Rightarrow Y \]

For bound variables \(X, Y\): \(Y\) is a consistent consequence of \(X\) iff \(\vdash X \Rightarrow Y\)
Example (All Work...)

\[
\begin{align*}
X_{s_0} &\overset{\nu}{=} Y_{s_0} & Y_{s_0} &\overset{\mu}{=} X_{s_2} \land X_{s_3} \land 
Y_{s_1} &\overset{\nu}{=} Y_{s_1} & Y_{s_1} &\overset{\mu}{=} Y_{s_0} 
Y_{s_2} &\overset{\nu}{=} Y_{s_2} & Y_{s_2} &\overset{\mu}{=} X_{s_1} 
Y_{s_3} &\overset{\nu}{=} Y_{s_3} & Y_{s_3} &\overset{\mu}{=} X_{s_2} 
\vdots & & \vdots & \end{align*}
\]

Is a consistent consequence of the BES:

\[
\begin{align*}
U_0 &\overset{\nu}{=} W_0 & W_0 &\overset{\mu}{=} U_2 
U_1 &\overset{\nu}{=} W_1 & W_1 &\overset{\mu}{=} W_0 
U_2 &\overset{\nu}{=} W_2 & W_2 &\overset{\mu}{=} U_1 \land U_2 
\end{align*}
\]

\[\vdash U_0 \Rightarrow X_{s_0} \text{ and } \vdash U_2 \Rightarrow X_{s_2} \text{ and } \vdash U_2 \Rightarrow X_{s_3} \text{ and } \ldots ;
\]

\[\vdash W_0 \Rightarrow Y_{s_0} \text{ and } \vdash W_2 \Rightarrow Y_{s_2} \text{ and } \vdash W_2 \Rightarrow Y_{s_3} \text{ and } \ldots ;
\]

e.g.: \[\vdash W_0 \Rightarrow Y_{s_0} \text{ follows if } W_0 \Rightarrow Y_{s_0} \vdash U_2 \Rightarrow X_{s_2} \land X_{s_3} \land \ldots \]
### Abstraction

#### Example (All Work...)

\[
\begin{align*}
X_{s_0} & \vdash Y_{s_0} \\
X_{s_1} & \vdash Y_{s_1} \\
X_{s_2} & \vdash Y_{s_2} \\
X_{s_3} & \vdash Y_{s_3} \\
\vdots & \\
Y_{s_0} & \equiv X_{s_2} \land X_{s_3} \land \ldots \\
Y_{s_1} & \equiv Y_{s_0} \\
Y_{s_2} & \equiv X_{s_0} \\
Y_{s_3} & \equiv X_{s_1} \\
\vdots & \\
\end{align*}
\]

Is a consistent consequence of the BES:

\[
\begin{align*}
U_0 & \vdash W_0 \\
U_1 & \vdash W_1 \\
U_2 & \vdash W_2 \\
W_0 & \equiv U_2 \\
W_1 & \equiv W_0 \\
W_2 & \equiv U_1 \land U_2 \\
\end{align*}
\]

e.g.: \( \vdash W_2 \Rightarrow Y_{s_2} \) follows if \( W_2 \Rightarrow Y_{s_2} \vdash U_1 \land U_2 \Rightarrow X_{s_1} \)

### Abstraction

#### Theorem (Soundness)

*If Y is a consistent consequence of X then X’s solution implies that of Y*

Abstraction works... but how well?

- Is there a comparable abstraction framework for transition systems?
- Which class of infinite BESs become potentially tractable to solve?
Abstraction

Generalised Kripke Modal Transition Systems

- May transitions
- Must hyper transitions

- Generalised mixed simulation

Theorem
Generalised Kripke Modal Transition Systems with generalised mixed simulation and BESs with consistent consequence are equally powerful for model checking

Theorem
Consistent consequence “abstractions” can be exponentially smaller than in generalised mixed simulation abstractions
Abstraction

Definition (Completeness for class $\mathcal{C}$ of BESs)
Consistent Consequence is complete for $\mathcal{C}$ iff for all $\mathcal{E} \in \mathcal{C}$

if equation $X \overset{\sigma}{=} f$ in $\mathcal{E}$ has solution $true$ for $X$ then
there must be a (finite) BES $\mathcal{E}'$ with equation $X' \overset{\sigma}{=} f'$ satisfying

- $\vdash X' \Rightarrow X$
- $X'$ has solution $true$

Theorem (Completeness Classes)
Consistent consequence is complete for:

- Greatest fixpoint-only (infinite) BESs
- Least fixpoint-only (infinite) BESs without infinite conjunctions

Outline

Fixpoint Logics for Verification

Boolean Equation Systems

Abstraction

Conclusions and Outlook
Consistent consequence is an abstraction framework for BESs
- coinductive generalisation of logical consequence
- abstractions remain within the same formalism
- theory extends to PBESs

Independent of application domain!
- model checking
- equivalence/simulation checking
- real-time model checking

Consistent consequence based tooling:
- abstraction using human-defined homomorphisms ........ pbesabsinthe
- predicate abstraction ........................................ future work
- CEGAR ....................................................... future work