Certificate Translation for Specification Preserving Advices

Gilles Barthe and César Kunz
INRIA Sophia Antipolis - Méditerranée

FOAL 2008
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
Local reasoning on:

- Baseline Code (to understand main functionality)
Local reasoning on:

- Baseline Code (to understand main functionality)

- Advice Code
  (to understand the implemented aspect)

Incremental concerns:
- Contract enforcement
- Logging / Profiling
- Evolving Security Requirements
Local reasoning on:

- Baseline Code (to understand main functionality)
- Advice Code (to understand the implemented aspect)

Incremental concerns:
- Contract enforcement
- Logging / Profiling
- Evolving Security Requirements

Global analysis of pointcuts to understand interaction of aspects
Producer vs Consumer Perspective

Obliviousness -> Local Reasoning?

Syntactic Obliviousness is not enough

Syntactic Obliviousness vs. Semantic Obliviousness

Dantas & Walker [POPL06]:
• characterize Harmless Advices that allow local reasoning
• information flow analysis to check advice non-interference.

PCC setting: contract enforcement
• functional properties (logic formulae)
• Absence of null pointer access
• Type Safety, etc.

Contract preserv. vs semantic preserv
weaker requirement

Satisfies contract P

Satisfies contract P
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
Specification Preserving Advices

\[
\{ x \geq 0 \}
\]

\[
\begin{align*}
& c := 1; \\
& x' := x; \\
& y' := y; \\
& \text{while } (y' \neq 1) \text{ do} \\
& \quad \text{if } (y' \mod 2 = 1) \text{ then} \\
& \quad \quad c := c.x' \\
& \quad \quad \text{fi} \\
& \quad \text{done;} \\
& x' = x'.c
\end{align*}
\]

\[
\{ x' = x^y \land y' = 1 \land c \geq 0 \}
\]

<table>
<thead>
<tr>
<th>Harmless</th>
<th>Spec. preserving</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Strong specification

\[
c := -5; \\
y' := 43
\]
Specification Preserving Advices

\[
\{ \textbf{True} \}
\]
\[
c := 1; \\
x' := x; \\
y' := y; \\
\text{while } (y' \neq 1) \text{ do} \\
\quad \text{if } (y' \mod 2 = 1) \text{ then} \\
\quad \quad c := c \cdot x' \\
\quad \text{fi} \\
\text{done;} \\
x' = x' \cdot c
\]

\[
\{ x' = x^y \}
\]

<table>
<thead>
<tr>
<th>Harmless</th>
<th>Spec. preserving</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

\[
c := -5; \\
y' := 43
\]
### Specification Preserving Advices

\[ \{ z = Z \} \]

\[
c := 1; \\
x' := x; \\
y' := y; \\
\text{while } (y' \neq 1) \text{ do} \\
  \text{if } (y' \mod 2 = 1) \text{ then} \\
    c := c.x' \\
  \text{fi} \\
\text{done}; \\
x' = x'.c
\]

\[ \{ x' = x^y \land z = Z \} \]

\[
c := -5; \\
y' := 43 \\
z := z + 1
\]

<table>
<thead>
<tr>
<th>Harmless</th>
<th>Spec. preserving</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
A specification preserving advice may modify variables in the specification.

<table>
<thead>
<tr>
<th>Harmless</th>
<th>Spec. preserving</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

- Output value may differ
- \( \text{in} \geq 0 \) is not invalidated.
- \( \text{even}(y) \) is ensured.
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
Proving spec-preservation

Baseline Code Verification: wp-based Vcgen

Verification of spec. preservation: wp-based Vcgen over modified advice code.
Proving spec-preservation

\[ \varphi \]

\[ \psi \]

\( W_f \)

... if b then
... x := proceed(e);
... fi
... 

\[ W_a \]

\[ \mathcal{V} \]

\[ W_f \]

\[ W_a \]

\[ W_a \cap \mathcal{V} \subseteq W_f \]
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
Certificate Translation for Specification Preserving Advices

Specification Harmless Advices

\[ \{ \phi_f \} \ f \ \{ \psi_f \} \]

\[ \{ \phi_g \} \ g \ \{ \psi_g \} \]

\[ \ldots \]
\[ \text{while } b \text{ do} \]
\[ \ldots \]
\[ \text{od} \]
\[ \ldots \]
\[ x := \text{proceed}(e); \]
\[ \ldots \]

\[ \phi_f \]
\[ \ldots \]
\[ \text{while } b \text{ do} \]
\[ \ldots \]
\[ \text{od} \]
\[ \ldots \]
\[ x := \text{call } f(e); \]
\[ \ldots \]

\[ \psi_f \]

\[ \phi_g \]
\[ \ldots \]
\[ \text{while } b \text{ do} \]
\[ \ldots \]
\[ \text{od} \]
\[ \ldots \]
\[ x := \text{proceed}(e); \]
\[ \ldots \]

\[ \psi_g \]
Certificate Translation for Specification Preserving Advices

Specification Harmless Advices

\[
x := \text{proceed}(e)
\]
\[
y := \text{proceed}(e')
\]
Specification Harmless Advices
Specification Harmless Advices

- Does not modify $V$
- $x := \text{proceed}(e)$
- $y := \text{proceed}(e')$
- Does not modify $V$ and $\text{res}=x$
Specification Harmless Advices

- Does not modify \( \mathcal{V} \)
- \( x := \text{proceed}(e) \)
- Does not modify \( \mathcal{V} \) and \( \text{res} = x \)

- \( y := \text{proceed}(e') \)

César Kunz (with Gilles Barthe)
Certificate Translation for Specification Preserving Advices

Specification Harmless Advices

\[
\begin{align*}
\phi & \quad \text{Does not modify } \mathcal{V} \\
\psi & \quad \text{Does not modify } \mathcal{V} \text{ and } \text{res}=x
\end{align*}
\]

\[
\begin{align*}
\phi & \quad \text{Does not modify } \mathcal{V} \\
\psi & \quad \text{Does not modify } \mathcal{V} \text{ and } \text{res}=x
\end{align*}
\]

\[
\begin{align*}
x := \text{proceed}(e) \\
y := \text{proceed}(e')
\end{align*}
\]
Certificate Translation for Specification Preserving Advices

Speciation Harmless Advices

\[\text{Does not modify } V\]

\[x := \text{proceed}(e)\]

\[\text{Does not modify } V\] and \(\text{res}=x\)

\[y := \text{proceed}(e')\]

\[\text{Does not modify } V\]
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
IMPROVING THE VERIFICATION POWER

\[ \varphi_f \]

\[ \psi_f \]

\[ x := \text{proceed}(e); \]

\[ \cdots \]

\[ x := \text{call } f(e); \]

\[ \cdots \]
IMPROVING THE VERIFICATION POWER

Certificate Translation for Specification Preserving Advices

\[ \varphi_f \]
\[ \psi_f \]

\[ \varphi_g \]
\[ \psi_g \]

\[ \varphi_f \]
\[ \psi_f \]

\[ \varphi_g \]
\[ \psi_g \]
Drawback

Multiple advised procedures = multiple verification invariants.

Or you want to verify the advice locally without considering for the moment in which contexts it will be executed!
IMPROVING THE VERIFICATION POWER

Drawback

Multiple advised procedures = multiple verification invariants.

(specification of proceed improves modularity)
Interference is not always a bad thing.

Some advices are be spec-preserving when combined but not when analyzed in isolation.
IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.

\[(\Phi_1, \Psi_1, \mathcal{W}_1)\]
\[a_1\]
\[(\Phi_1^p, \Psi_1^p, \mathcal{W}_1^p)\]

\[(\Phi_2, \Psi_2, \mathcal{W}_2)\]
\[a_2\]
\[(\Phi_2^p, \Psi_2^p, \mathcal{W}_2^p)\]

\[\ldots\]

\[(\Phi_n, \Psi_n, \mathcal{W}_n)\]
\[a_n\]
\[(\Phi_n^p, \Psi_n^p, \mathcal{W}_n^p)\]
Certificate Translation for Specification Preserving Advices

IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.

\[(\Phi_1, \Psi_1, \mathcal{W}_1)\]
\[a_1\]
\[(\Phi_1^P, \Psi_1^P, \mathcal{W}_1^P)\]

\[(\Phi_2, \Psi_2, \mathcal{W}_2)\]
\[a_2\]
\[(\Phi_2^P, \Psi_2^P, \mathcal{W}_2^P)\]

... 

\[(\Phi_n, \Psi_n, \mathcal{W}_n)\]
\[a_n\]
\[(\Phi_n^P, \Psi_n^P, \mathcal{W}_n^P)\]

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.
IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]

Baseline proc.

\[(\Phi_1, \Psi_1, \mathcal{W}_1)\]

\[a_1\]

\[(\Phi_1^P, \Psi_1^P, \mathcal{W}_1^P)\]

\[(\Phi_2, \Psi_2, \mathcal{W}_2)\]

\[a_2\]

\[(\Phi_2^P, \Psi_2^P, \mathcal{W}_2^P)\]

\[\ldots\]

\[(\Phi_n, \Psi_n, \mathcal{W}_n)\]

\[a_n\]

\[(\Phi_n^P, \Psi_n^P, \mathcal{W}_n^P)\]

\[(\Phi', \Psi', \mathcal{W}')\]

\[a_1\]

\[(\Phi, \Psi, \mathcal{W})\]

Baseline proc.
Certificate Translation for Specification Preserving Advices

IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.

\[\begin{array}{l}
(\Phi_1, \Psi_1, \mathcal{W}_1) \\
(\Phi_1^P, \Psi_1^P, \mathcal{W}_1^P) \\
a_1
\end{array} \quad \begin{array}{l}
(\Phi_2, \Psi_2, \mathcal{W}_2) \\
(\Phi_2^P, \Psi_2^P, \mathcal{W}_2^P) \\
a_2
\end{array} \quad \ldots \quad \begin{array}{l}
(\Phi_n, \Psi_n, \mathcal{W}_n) \\
(\Phi_n^P, \Psi_n^P, \mathcal{W}_n^P) \\
a_n
\end{array} \]

\[(\Phi'', \Psi'', \mathcal{W}'') \]

\[\begin{array}{l}
(\Phi', \Psi', \mathcal{W}') \\
a_2
\end{array} \quad \begin{array}{l}
(\Phi, \Psi, \mathcal{W}) \\
a_1
\end{array} \quad \text{Baseline proc.}\]
Certificate Translation for Specification Preserving Advices

IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.

\[(\Phi_1, \Psi_1, \mathcal{W}_1)\]
\[a_1\]
\[(\Phi_P^1, \Psi_P^1, \mathcal{W}_P^1)\]

\[(\Phi_2, \Psi_2, \mathcal{W}_2)\]
\[a_2\]
\[(\Phi_P^2, \Psi_P^2, \mathcal{W}_P^2)\]

\[
\text{...}\]

\[(\Phi_n, \Psi_n, \mathcal{W}_n)\]
\[a_n\]
\[(\Phi_P^n, \Psi_P^n, \mathcal{W}_P^n)\]

\[
\text{...}\]

\[(\Phi'', \Psi'', \mathcal{W}'')\]

\[
\text{...}\]

\[(\Phi', \Psi', \mathcal{W}')\]
\[a_2\]

\[
\text{...}\]

\[(\Phi, \Psi, \mathcal{W})\]
\[a_1\]

Baseline proc.

César Kunz (with Gilles Barthe)
IMPROVING THE VERIFICATION POWER

\[(\Phi, \Psi, \mathcal{W})\]
Baseline proc.

\[(\Phi_1, \Psi_1, \mathcal{W}_1)\], \(a_1\)

\[(\Phi^p_1, \Psi^p_1, \mathcal{W}^p_1)\]

\[(\Phi_2, \Psi_2, \mathcal{W}_2)\], \(a_2\)

\[(\Phi^p_2, \Psi^p_2, \mathcal{W}^p_2)\]

... \(a_n\)

\[(\Phi_n, \Psi_n, \mathcal{W}_n)\]

\[(\Phi^p_n, \Psi^p_n, \mathcal{W}^p_n)\]
\( \Phi \Rightarrow \Phi^n \quad \Psi^n \Rightarrow \Psi \quad \mathcal{W}^n \cap \mathcal{V} \subseteq \mathcal{W} \)

Specification Refinement instead of Specification Preservation

Baseline proc.
MOTIVATION

SPECIFICATION PRESERVING ADVICES

PROVING SPECIFICATION PRESERVING ADVICES

REDUCING PROOF OBLIGATIONS

IMPROVING THE VERIFICATION POWER

CERTIFICATE TRANSLATION
Certificate Translation

Certificate Translation for Specification Preserving Advices

Producer

Source Program → Compiler → Compiled Program → Execution

VCGen → Verification Conditions → Prover

Certificate Translator → Certificate

Consumer

VCGen → Verification Conditions

Certificate → Proof Checker → OK
Certificate Translation

Consider the situation:
- Client verification and execution environment not AOP-oriented
- Code generated by multiple producers is weaved before execution

![Diagram showing the certificate translation process and the relationship between code producers and the execution environment.](Diagram.png)
Certificate Translation

Consider the situation:

- Client verification and execution environment not AOP-oriented
- Code generated by multiple producers is weaved before execution
Certificate Translation

High level/structured

Baseline Source Code

Compiler

Low level/stack based

Low level Code
Certificate Translation

High level/structured

Baseline Source Code

Certificate Translation

(\phi, \psi)

Certificate

Low level/structured

Compiler

Low level Code

(\phi, \psi)

Certificate

César Kunz (with Gilles Barthe)
Certificate Translation

High level/structured

- Baseline Source Code
- Advice Source Code
- \((\phi, \psi)\) Certificate

Low level/stack based

- Compiler + Weaving
- Final Weaved Code
- \((\phi, \psi)\) Certificate

Certificate Translation for Specification Preserving Advices
Certificate Translation

High level/structured

Baseline Source Code

Advice Source Code

$(\phi, \psi)$ Certificate

$(\phi, \psi)$-preserv. Certificate

Low level/stack based

Compiler

Compiler + Weaving

Certificate Translation

Cert. Trans. for adv. weaving

Low level Code

Final Weaved Code

$(\phi, \psi)$ Certificate

$(\phi, \psi)$ Certificate
Conclusions

- A more flexible notion of non-interfering advices
- Stronger non-interference analyses reduce proof obligations
- Certificate translation targetting a typical backend