Composing Modal Properties of Programs with Procedures

Dilian Gurov

Royal Institute of Technology, Stockholm

Marieke Huisman

INRIA Sophia-Antipolis

FESCA 2007, Braga

March 24, 2007

Overview

1. Motivation
2. A Framework for Compositional Verification
3. From Behavioural to Structural Properties
4. Correctness of Translation
5. Conclusions and Future Work

1. Motivation: Security of Mobile Applications

Small secure devices (*e.g.* smart cards)

- store privacy–sensitive data
- require strong guarantees of security: formal verification

Interacting applications (*e.g.* JavaCard applets)

- communication via method invocation over shared interfaces
- example: electronic purse applet and several loyalties

Dynamic loading (post–issuance)

- ability to load new applets after the device has been put in operation
- requires compositional verification

Compositional Verification

Compositional Verification Principle

$$\frac{\models A : \psi \qquad X : \psi \models X \otimes B : \phi}{\models A \otimes B : \phi}$$

premises: local property of A and correctness of decomposition

Scenarios for secure post-issuance loading

1. device issuer specifies ϕ and ψ and checks property decomposition;
   pre-load check of $\models A : \psi$
2. device issuer provides only ϕ, applet provider specifies ψ;
   pre-load check of $\models A : \psi$ and property decomposition

Maximal Models for Compositional Verification

In certain setups

- property preserving simulation preorder
- for any formula ψ, the set of models for ψ has a maximal element $Max(\psi)$ w.r.t. the preorder: maximal model
- simulation preorder preserved by composition $\otimes$

Maximal Model Principle [Grumberg & Long '94]

$$\frac{\models Max(\psi) \otimes B : \phi}{X : \psi \models X \otimes B : \phi}$$

- Derived Compositional Verification Principle

$$\frac{\models A : \psi \qquad \models Max(\psi) \otimes B : \phi}{\models A \otimes B : \phi}$$

  - premises: local property of A and correctness of decomposition
  - now: pure model checking
  - but: requires maximal model construction (expensive)

Previous Work

Theory [Sprenger, Huisman, Gurov: MEMOCODE'04]

- formal framework
- maximal model construction
- sound and complete composition rule

Case Study [Huisman, Gurov, Sprenger, Chugunov: FASE'04]

- electronic purse with loyalty programmes
- by smart card provider Gemplus
- verified absence of illicit applet interactions

Present Paper

Summary

- characterise behaviour through structure
- via translation from behavioural properties to structural ones
- extends above method to local behavioural properties

2. Framework for Compositional Verification

Model Labelled transition system + Valuation

Simulation Preorder $\leq$ standard definition

Simulation Logic modal logic with box modalities and gfp recursion:

$$\phi ::= p \mid \neg p \mid X \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid [a]\,\phi \mid \nu X.\phi$$

Maximal Models $Max(\psi)$

- exist
- construction: exponential, lazy

Applet Structure

Applet $\mathcal{A}$

- control–flow graph represented as model
- applet composition $\uplus$
- structural simulation and properties

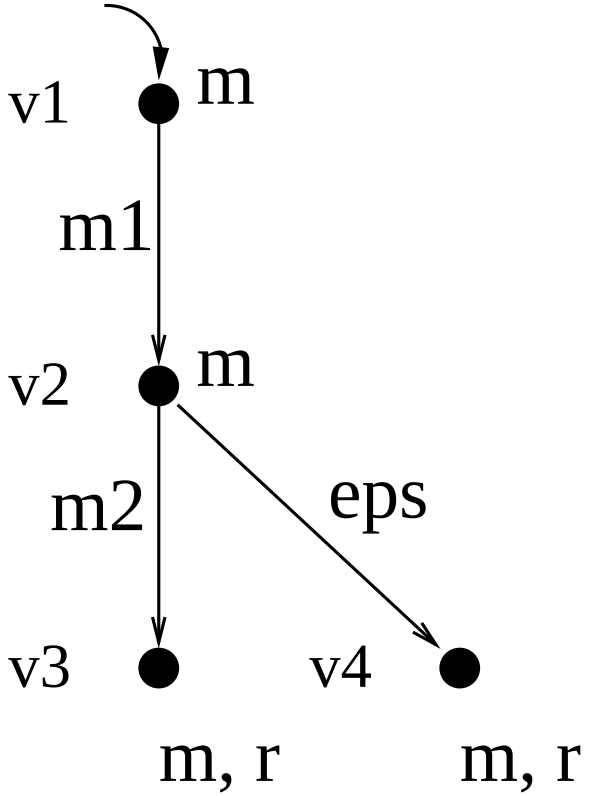
Maximal Model for property ψ is not necessarily a legal applet structure!

- interface $I = (I^+, I^-)$ of provided and required methods
- formula ϕ_I axiomatizing applets with interface I

Maximal Applet $Max_I(\psi)$

- is the maximal model $Max(\phi_I \wedge \psi)$

Example Method Graph



Applet Behaviour

- Applet structure $\mathcal{A}$ induces applet behaviour $b(\mathcal{A})$

  - configurations: pairs (v, σ) of control point and call stack

  - labels: ε, $m_1\, call\, m_2$, $m_2\, ret\, m_1$

  - transitions: standard, induced in a context–free manner

- Behavioural simulation and properties

  - applet interaction properties

- Applet behaviour is not axiomatizable within the logic...

  ...but (at least) structural simulation implies behavioural simulation!

Operational Semantics

[transfer] $(v, \sigma) \xrightarrow{\tau}_b (v', \sigma)$ if $v \xrightarrow{\varepsilon}_m v', v \models \neg r$

[call] $(v_1, \sigma) \xrightarrow{m_1 \text{ call } m_2}_b (v_2, v_1' \cdot \sigma)$ if $v_1 \xrightarrow{m_2}_{m_1} v_1', v_1 \models \neg r,$ $v_2 \models m_2, v_2 \in E$

[return] $(v_2, v_1 \cdot \sigma) \xrightarrow{m_2 \text{ ret } m_1}_b (v_1, \sigma)$ if $v_2 \models m_2 \wedge r, v_1 \models m_1$

Compositional Verification Method

Compositional Verification Principle

$$\frac{\mathcal{A} \models_s \sigma \qquad \mathcal{M}ax_{I_{\mathcal{A}}}(\sigma) \uplus \mathcal{B} \models_b \psi}{\mathcal{A} \uplus \mathcal{B} \models_b \psi} \; \mathcal{A} : I_{\mathcal{A}}$$

1. a) Specify global property ψ as a behavioural property

   b) For applet $\mathcal{A}$, specify local property σ as a structural property

2. Verify the correctness of the property decomposition:

   a) compute maximal applet $\mathcal{M}ax_{I_{\mathcal{A}}}(\sigma)$

   b) model check $\mathcal{M}ax_{I_{\mathcal{A}}}(\sigma) \uplus \mathcal{B} \models_b \psi$

3. When implementation of $\mathcal{A}$ available, verify $\mathcal{A} \models_s \sigma$

Structural vs Behavioural Properties

Structural properties are

- less abstract
- but far more efficient to verify!

Present Paper

- characterise behaviour through structure
- via translation from behavioural properties to structural ones
- extending the above method to local behavioural properties

3. From Behavioural to Structural Properties

Problem

- in general: no unique maximal applet for behavioural properties

Example

- behavioural property: $[a\ \mathsf{call}\ b]\, r$
- structural property: $a \Rightarrow [b]\,\mathsf{ff}$
- structural property: $b \Rightarrow r$

Idea

- characterise behavioural properties through sets of structural ones

The Translation

Idea

- symbolic execution of behavioural formula
- accumulate structural constraints
- by means of history stack: $(m, F) \cdot H$

Translation

- for modal fragment: simple mapping π_H
  and define $\Pi_I(\phi) = \{\bigwedge_{m \in I^+} \sigma_m \mid \sigma_m \in \pi_{(m,\epsilon)}(\phi)\}$
- for full logic: involved tableau construction

The Mapping π_H

$$
\begin{array}{rcl}
\pi_{(i,F)\cdot H}(p) & = & \{i \Rightarrow [F]\, p\} \cup \{i' \Rightarrow \left[F'\right] \mathsf{ff} \mid (i', F') \in H\} \\
\pi_{(i,F)\cdot H}(\neg p) & = & \{i \Rightarrow [F]\, \neg p\} \cup \{i' \Rightarrow \left[F'\right] \mathsf{ff} \mid (i', F') \in H\} \\
\pi_{(i,F)\cdot H}(\phi_1 \wedge \phi_2) & = & \{\sigma_1 \wedge \sigma_2 \mid \sigma_1 \in \pi_{(i,F)\cdot H}(\phi_1),\ \sigma_2 \in \pi_{(i,F)\cdot H}(\phi_2)\} \\
\pi_{(i,F)\cdot H}(\phi_1 \vee \phi_2) & = & \pi_{(i,F)\cdot H}(\phi_1) \cup \pi_{(i,F)\cdot H}(\phi_2) \\
\pi_{(i,F)\cdot H}([\tau]\, \phi) & = & \pi_{(i,F\cdot\varepsilon)\cdot H}(\phi) \\
\pi_{(i,F)\cdot H}([a\ \mathsf{call}\ b]\, \phi) & = & \begin{cases} \{\mathsf{tt}\} & \text{if } i \neq a \\ \pi_{(b,\epsilon)\cdot(i,F\cdot b)\cdot H}(\phi) & \text{if } i = a \end{cases} \\
\pi_{(i,F)\cdot H}([a\ \mathsf{ret}\ b]\, \phi) & = & \begin{cases} \{\mathsf{tt}\} & \text{if } i \neq a \vee \ldots \\ \{i \Rightarrow [F]\, \neg r\} \cup \pi_H(\phi) & \text{if } i = a \wedge \ldots \end{cases}
\end{array}
$$

Examples

Example1

$$\begin{aligned} \pi_{(a,\epsilon)}([a\ \mathsf{call}\ b]\, r) &= \pi_{(b,\epsilon)\cdot(a,b)}(r) \\ &= \{b \Rightarrow r, a \Rightarrow [b]\,\mathsf{ff}\} \end{aligned}$$

Example2

$$\begin{aligned} \pi_{(a,\epsilon)}([a\ \mathsf{call}\ b]\,[a\ \mathsf{call}\ b]\, r) &= \pi_{(b,\epsilon)\cdot(a,b)}([a\ \mathsf{call}\ b]\, r) \\ &= \{\mathsf{tt}\} \end{aligned}$$

4. Correctness of Translation

Definition [Generalized Satisfaction]

$$\mathcal{A} \models_H \phi \Leftrightarrow \forall v, \sigma.(\gamma_{\mathcal{A}}(v \cdot \sigma, H) \Rightarrow (v, \sigma) \models_b \phi)$$

Proposition

$$\mathcal{A} \models_b \phi \Leftrightarrow \forall m \in I^+ . \mathcal{A} \models_{(m,\epsilon)} \phi$$

Theorem Let ϕ be disjunction-free. Then:

$$\mathcal{A} \models_H \phi \Leftrightarrow \exists \sigma \in \pi_H(\phi). \mathcal{A} \models_s \sigma$$

Compositional Verification Method

Compositional Verification Principle

$$\frac{\mathcal{A} \models_b \phi \qquad \{\mathcal{M}ax_{I_\mathcal{A}}(\sigma) \uplus \mathcal{B} \models_b \psi\}_{\sigma \in \Pi(\phi)}}{\mathcal{A} \uplus \mathcal{B} \models_b \psi} \; \mathcal{A} : I_\mathcal{A}$$

1. a) Specify global property ψ as a behavioural property

   b) For applet $\mathcal{A}$, specify local property ϕ as a behavioural property

2. Verify the correctness of the property decomposition: for all $\sigma \in \Pi(\phi)$

   a) compute maximal applet $\mathcal{M}ax_{I_\mathcal{A}}(\sigma)$

   b) model check $\mathcal{M}ax_{I_\mathcal{A}}(\sigma) \uplus \mathcal{B} \models_b \psi$

3. When implementation of $\mathcal{A}$ available, verify $\mathcal{A} \models_b \phi$

5. Conclusion

We presented:

- a translation from behavioural to structural properties
- with correctness proof

Benefits:

- extends compositional verification method:
  support for local behavioural properties
- independent value: relationship structure vs behaviour

Future work

- translation for full simulation logic: tableau construction

New Slide

text