Formal Methods Research at SICS and KTH
- An Overview -

Mads Dam
SICS and KTH/IMIT

Executive Summary

- Group of researchers from SICS and IMIT, KTH
- ~ 3 researchers and 3 PhD students plus visitors
- Research theme: Software security
  - Program analysis + verification techniques for security in broad sense
  - Compositional verification and mu-calculus ←
  - Security protocol verification
  - Information flow theory and practice ←
  - Authorisation, PKI, and policy-based management
  - JavaCard verification ←
- Funding from: EU, Ericsson, Microsoft, USAF, Vinnova

I: First-Order Mu-Calculus As a Framework for Program Verification

What is a good framework for verification of first-order (distributed) programs?

Hoare logic? Too messy and ad-hoc

HO type theory? Too general by far

Model checking? Not for "general" programs

Needed:

First-order logic + induction + coinduction

= first-order mu-calculus

Approach

Basis:

- Gentzen-type proof system for FOMuC
- Explicit ordinal approximations
- Loop discharge mechanism for automatically resolving nested inductions/coinductions !

Language embedding:

- Induction + data type constructors:
- Data types: Nat = mu X(n).n=0 \/ exists n1.n=n1+1 ...
- Language: Prog = mu X(p).p=skip \/ exists p1,p2. ...
- States:

  State = \s. (exists p,t.Prog(p) /\ Store(t) /\ s = (p,t)) \/ ...
- Embeddings of operational semantics:

TransRel =

mu X(s1,s2).(exists t.Store(t) /\ s1=(skip,t) /\ s2 = t) \/ ...

Results

Theorem-proving basics:

- Ordinal approximations, soundness and completeness of discharge (Dam, Gurov, Sprenger)

Language embedding framework:

- General, compositional verification (Simpson-95,Dam-95,Fredlund-01)
- Instantiations – CCS, Erlang, pi-calculus, JavaCard (Papers by Dam, Fredlund, Gurov, Chugunov a.o.)
- Completeness for context-free + pushdown cases (Simpson-Schoepp)

Case studies

- Erlang (Arts-Dam), JavaCard (Huisman-Gurov-Barthe)

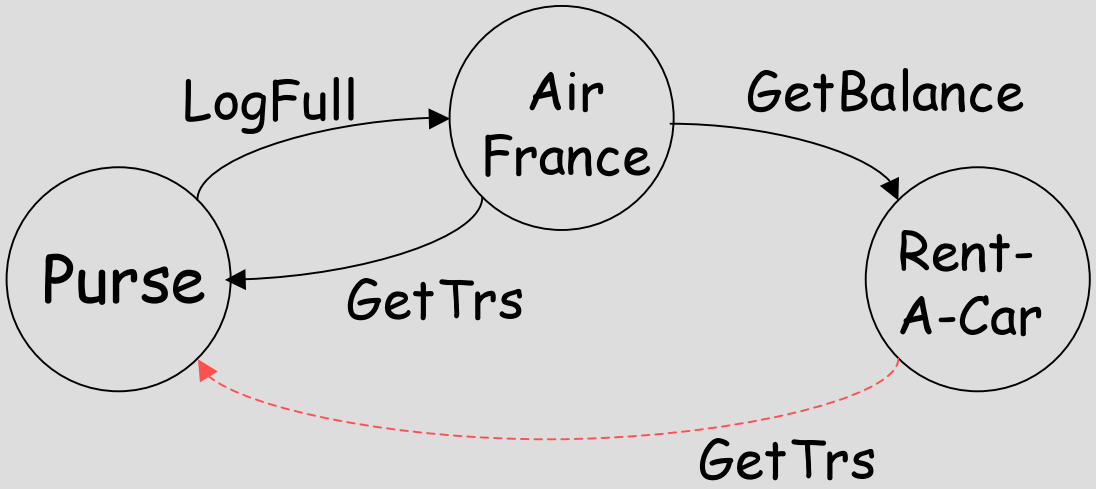
Tools

- www.sics.se/fdt/vericode (Fredlund)

II: JavaCard Applet Interaction

Fine-grained control of applet interaction is sometimes needed

Example (Gemplus, PACAP):



Rent-A-Car may not subscribe to LogFull, but may acquire information indirectly

Approach and Results

Multi-applet control-flow property:

- Does call to LogFull cause call of GetTrs by Rent-A-Car?

Applets modelled as pushdown automata

Desired property modelled using LTL:

- Is there a call to GetTrs between call and exit of LogFull?

Compositional verification reduce global checks to per-applet checks – for post-issuance loading

Papers by Huisman, Gurov, Barthe, Fredlund, Chugunov, Sprenger

Toolset in progress

III: Information Flow Control

How to protect against side channels in the presence of cryptography and/or explicit downgrading?

Applets which perform/use

- Crypto and crypto-related op's
- Key and pin management
- Initialisation/deletion/recovery/update op's
- Access+authorisation control

Multi-level security model not applicable

- There is flow of information

Approach and Results

Admissible interference:

- Specify intended information flow
- Check that no other channels exist

Semantics: Invariance of behaviour under replacement of secrets

Volpano-Smith-like condition:

- If applet respects flow spec and no "branching on Hi" then admissibility holds

Papers by Giambiagi and Dam

In pipeline: Adaptation to JavaCard, analyzer, case studies