Compositional Verification of Binary Code

Didrik Lundberg

Supervised by Mads Dam and Roberto Guanciale

Example HolBA Verification Workflow

Input
- Binary is read into HOL4
- Interpreted using an exported L3 or Sail ISA model
- Contract: $\{P_{bin}\}\ C_{bin}\ \{Q_{bin}\}$

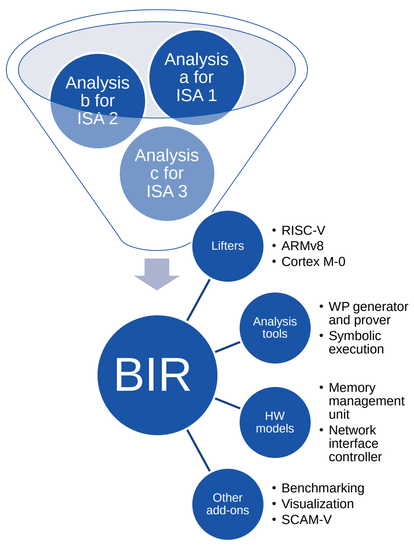
Lift binary to BIR
- Yields lifter theorem
- Contract is translated to BIR: $\{P_{BIR}\}\ C_{BIR}\ \{Q_{BIR}\}$

Generate VC in BIR
- Prove contract with weakest precondition WP_{BIR}
- Use Rule of Consequence for precondition strengthening

Prove VC: $P_{BIR} \Rightarrow WP_{BIR}$
- Using SMT solver (Z3)
- Yields BIR contract

Prove binary contract
- Using backlifting theorem, BIR contract and lifter theorem
- Benefits from unified transition system

TCB: ISA model, SMT solver, HOL4 logic



HolBA enables code re-usability and maintainability for more effective use of programmer time

In previous work, composition of contracts:	× Is ambiguous on the number of execution steps [Kumar 2006]
	× Can't handle overlapping code [Saabas 2006]
	× Can't merge variable contexts
	× Requires passing around large constructs [Tan 2006]
	× Can't handle total correctness [Tan 2006]
	× Only handles partial correctness [Saabas 2006]

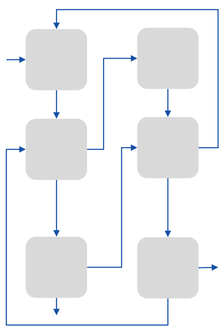
Our work introduces compositional contracts with none of these drawbacks, which enable an incremental approach to verification of unstructured code

The goal for the future is to apply the new capacities of HolBA to verify larger and more complex programs, for example TEE frameworks. Another possible usage would be as translation validation between Cogent or F* and binary.

SCAN ME

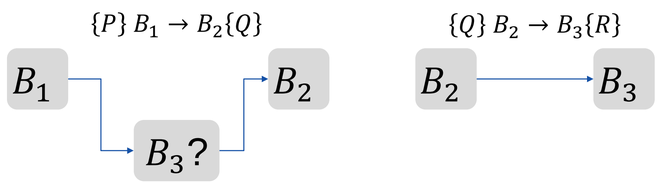
This effort has resulted in new HOL4 theories for weak transition systems and their instantiations in HolBA for both binary and BIR

Problem



Unstructured code includes any control flow pattern: loops, function calls, mutual recursion between an arbitrary amount of functions, …

Example



HT definition: Q is established once B2 is reached
Problem: Possibly encountering end point of second contract inside first contract

Candidate solution: Change HT to say that $\exists n$ such that n-block execution reaches end point and establishes Q.
But then:

$$\{P\}\ B_1 \rightarrow B_2\{Q_1\}\ \wedge \{P\}\ B_1 \rightarrow B_2\{Q_2\}\ \not\Rightarrow \{P\}\ B_1 \rightarrow B_2\{Q_1 \wedge Q_2\}$$

Improved solution: Introduce blacklist of non-visited blocks.

References

TAN, G; APPEL, A W. 2006. A compositional logic for control flow. *International Workshop on Verification, Model Checking, and Abstract Interpretation*. p. 80-94.

SAABAS, A; UUSTALU, T. 2006 A compositional natural semantics and Hoare logic for low-level languages. *Electronic Notes in Theoretical Computer Science*, p. 151-168.

KUMAR, R; et al. CakeML: a verified implementation of ML. *ACM SIGPLAN Notices*, p. 179-191.

FUNDED BY:

Stiftelsen för Strategisk Forskning

SAAB Technologies

SAAB

KTH ROYAL INSTITUTE OF TECHNOLOGY

Didrik Lundberg

Lindstedtsvägen 3
114 28 Stockholm

Email: didrikl@kth.se