Quantitative Contract-Based Synthesis and Verification for CPS Security

Alberto Sangiovanni-Vincentelli, Sanjit Seshia
University of California, Berkeley

The problem

- Industrial cyber-physical systems (CPS) are being increasingly designed using reusable components and “plug-and-play” architectures.
- This approach to design brings renewed concerns about security and privacy, and about balancing them with other concerns, such as safety.
- We want to reason about security using assume-guarantee contracts to maintain the compositionality of the design. However, AG contracts cannot express security attributes.

Assume-guarantee contracts

- A contract *C = (A,G)* consists of sets *A* and *G* of *behaviors* which are *assumptions* and *guarantees*, respectively, for the component.
- Contracts have a partial order called ***refinement***. Contract *C’ = (A’, G’)* refines *C*, written *C’* ≤ *C* iff *G’* ⊆ *G* and *A* ⊆ *A’*. Refinement is related to substitutability: a contract can be replaced with a refinement without altering the functionality of a design.
- Contracts have a binary operation called ***composition***, which yields the the specification of a system composed of elements adhering to the contracts being composed.
- Given a top-level specification *C = (A,G)*, and the specification *C’=(A’, G’)* of a component to be used in the design, the least-stringent specification of a component, *C/C’*, that composed with *C’* refines the top-level spec is called ***quotient***. This operation is the adjoint of composition.
- Contracts have another operation, called ***merging***, which allows us to generate a single specification for a design element starting with specifications that describe multiple viewpoints of the design element.
- Merging also has an adjoint operation, called ***separation***.

Composition

$C_1 \parallel C_2 = ((A_1 \cap A_2) \cup \neg(G_1 \cap G_2), G_1 \cap G_2)$

Merging

$C_1 \bullet C_2 = (A_1 \cap A_2, \quad (G_1 \cap G_2) \cup \neg(A_1 \cap A_2))$

Quotient

$C / C' = (A \cap G', \; (G \cap A') \cup \neg(A \cap G'))$

Separation

$C \div C' = ((A \cap G') \cup \neg(G \cap A'), \quad G \cap A')$

$$\begin{tikzcd}
C \arrow[r, dashed, mapsto, "() / C'"] & C / C' \\
C'' \parallel C' \arrow[u] & C'' \arrow[l, dashed, mapsto, "() \parallel C'"'] \arrow[u]
\end{tikzcd}$$

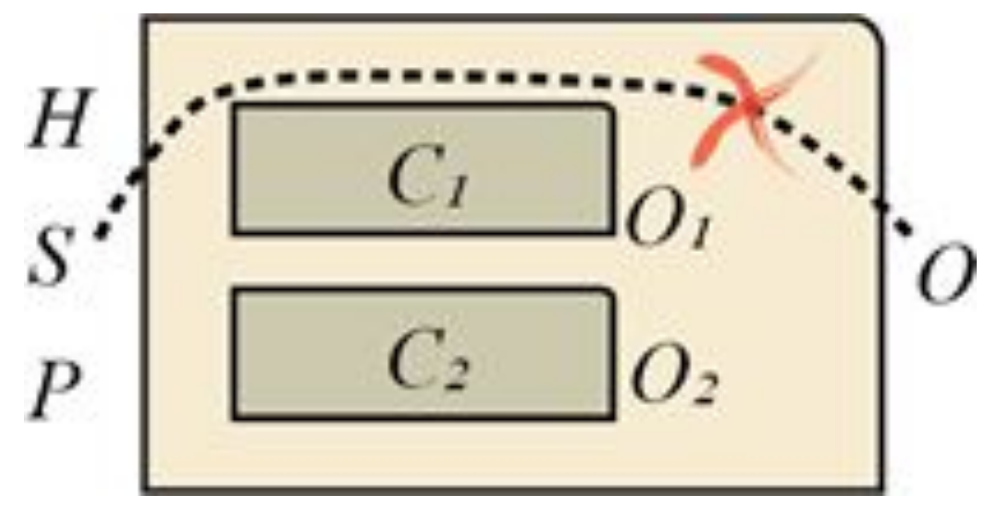
$$\begin{tikzcd}
C \arrow[r, dashed, mapsto, "() \div C'"] \arrow[d] & C \div C' \arrow[d] \\
C'' \bullet C' & C'' \arrow[l, dashed, mapsto, "() \bullet C'"']
\end{tikzcd}$$

Incer et al., Quotient for Assume-Guarantee Contracts. MEMOCODE 18.
Passerone et al., Coherent Extension, Composition, and Merging Operators in Contract Models for System Design. EMSOFT 19.
Incer et al., The Quotient in Preorder Theories. GandALF 2020.

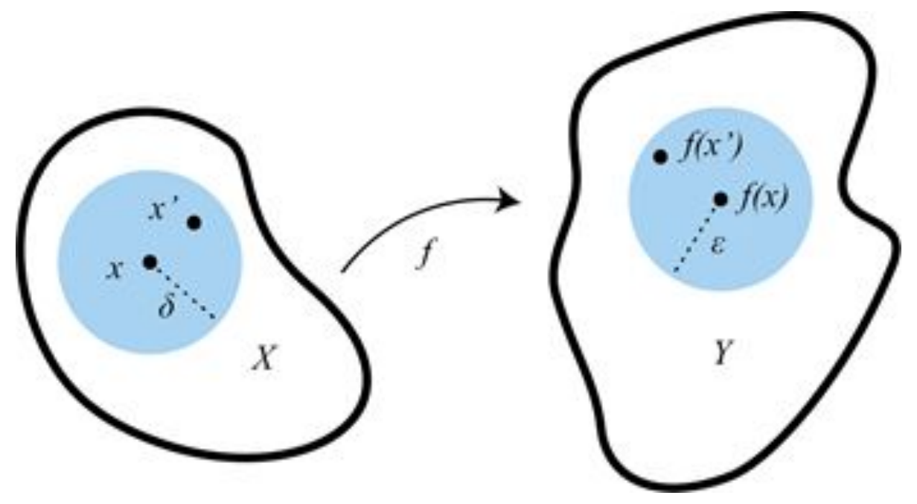
Approach

- Develop a contract-based theoretical framework that enables compositional reasoning about security properties of a system.
- This requires extending assume-guarantee contracts to deal with hyperproperties, which can express information-flow constraints and non-interference, important for security.

Scientific Impact



The algebra of assume-guarantee contracts is now complete and can be used to reason compositionally about any system attribute that can be expressed as a trace property.



We are also developing a theory of hypercontracts, which enables compositional reasoning of any system attribute. In particular, it allows expressing information-flow security properties and the robustness of data-driven components.

Impact on Society

Our work

- Supports “plug-and-play” methodologies.
- Formalizes component specifications and aids the interaction of an OEM with its suppliers.
- Helps system designers identify exactly the specifications that need to be implemented in a design.
- Yields faster design of safer and more secure automobiles, planes, factories, etc.

Impact on Education

Our theory is an important contribution to contract-based design. It is taught in embedded design courses such as EE249B at UC Berkeley.

2021 NSF Cyber-Physical Systems Principal Investigators' Meeting
June 2-4, 2021

Award ID#: 1739816