Decomposing Specifications Using the Quotient of Assume-Guarantee Contracts

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

Berkeley
UNIVERSITY OF CALIFORNIA

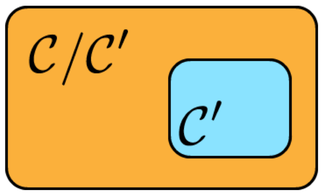
What are Assume-Guarantee Contracts?

- 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 $C' \leq C$, iff $G' \subseteq G$ and $A \subseteq A'$. 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 represents the concurrent operation of two contracts. It is given by

$$\mathcal{C} \otimes \mathcal{C}' = (A \cap A' \cup \neg G \cup \neg G', G \cap G')$$

The Problem

- Given a top-level specification *C = (A,G)*, and the specification *C'=(A', G')* of a component to be used in the design, we want the least-stringent specification of a component, *C/C'*, that composed with *C'* refines the top-level spec.
- The quotient identifies the "missing specification" *C/C'* in the design.

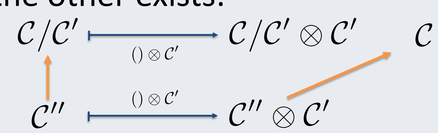


Solution

The quotient of assume-guarantee contracts is defined as a first-order expression:

$$\forall \mathcal{C}''.\ \mathcal{C}'' \otimes \mathcal{C}' \leq \mathcal{C} \Leftrightarrow \mathcal{C}'' \leq \mathcal{C}/\mathcal{C}'$$

That is, a contract *C''* is a solution to our problem iff it refines the quotient *C/C'*. If we represent refinement with arrows ⟶, in the following diagram the quotient is the element with the universal property that one of the arrows ⟶ exists iff the other exists:

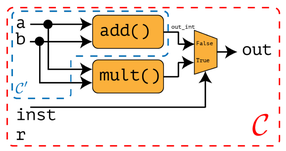


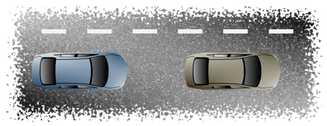
Before our work, a closed form expression for the quotient was not known. We proved that the quotient of assume-guarantee contracts is given by

$$\mathcal{C}/\mathcal{C}' = (A \cap G', A' \cap G \cup \neg(A \cap G'))$$

Scientific Impact

The quotient operation is useful in *any application* that seeks to identify functionality that still needs to be implemented. Our example use-cases include finding the specs of missing functionality for an ALU, and for the vehicle network used in a cooperative collision avoidance application.





Among other applications, we have distributed control, requirement engineering, and controller synthesis.

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 will be taught in embedded design courses at UC Berkeley.

CPS: Medium: Quantitative Contract-Based Synthesis and Verification for CPS Security

Award ID#: 1739816