Formal Verification of Distributed Algorithms

K. Mani Chandy¹  Brian Go¹  Sayan Mitra²  Jerome White¹

¹California Institute of Technology
²University of Illinois at Urbana-Champaign

Overview

- Improve distributed software reliability
- Ease proof burden for concrete algorithms
- Stepwise refinement and program transformation
- Organize library of theorems for distributed systems (PVS)
- Transform theorem proven code into executable code (Java/Erlang)
- Improve teaching

Abstractions

Incorporate fundamental abstractions of distributed algorithms that are generic enough to be applicable to a large class of problems

Fundamental Functions

- Binary operator \( \circ : T \to T \)
- Composition function \( f : S, \{ K \subseteq S \}, \circ \to T \)
- Transition Predicate \( t : S, S \to \text{Bool} \)

Correctness Properties

- Safety: invariant function with respect to start state \( t(S^0, S) \Rightarrow f(S, \{ K \subseteq S \}, \circ) = f(S^0, \{ K = S \}, \circ) \)
- Progress: variant function to a well founded set \( \rho : T \to T | T \text{ is well founded} \)

Application: Consensus

Abstractions

- Given state \( S \) with \( n \) agents, want final state to be function of start state \( S^* : V \in S : S^*(v) = f(S) \)
- \( \circ \) commutative \& associative \& (idempotent \& super idempotent)
- fold \( (f) \) is aggregation \( (\text{fold}) \) of composable binary operator
- \( f \) is a depth first graph traversal method

Visual Transition

- \( S \)
- \( a, b, c, d \)
- \( S^0 \)
- \( S' \)
- \( S^0 \Rightarrow f(S) = f(S') \)
- \( a' \), \( b' \), \( c' \), \( d' \)
- \( \circ \)

Object Refinement

- \( \circ (S, S) \to T \)
- \( \text{fold}(S, J, \circ) \to T \)
- \( \text{FoldableOperator} \)
- \( \min / \max \), \( \text{gcd/lcm} \), \( \text{average} \), \( \text{convex hull} \)

Proof Refinement

- \( \circ (S, S) \to T \)
- \( \text{fold}(S, J, \circ) \to T \)
- \( \text{commutative, associative} \)

Application: Graphs

Abstractions

- Given source graph \( G \) and root vertex \( R \), produce target graph that solves given algebraic path problem
- \( \circ \) is semiring with additional constraints
- \( \circ : S^0 \to S' \)
- \( (G, \oplus, \otimes, 0, 1) \)

Visual Transition

- \( (\{ P \}, \oplus, \otimes, 0, 1) \)

Object Refinement

- \( \circ (\{ P \}, \oplus, \otimes, 0, 1) \to T \)
- \( \text{GraphTraverser} \)
- \( \text{Reachability} \)
- \( \text{ShortestPath} \)

Proof Refinement

- \( \circ (\{ P \}, \oplus, \otimes, 0, 1) \to T \)
- \( \text{reachability} \)
- \( \text{shortest path} \)

References
