Incorrectness Specification Inference

ANONYMOUS AUTHOR(S)

1 PROBLEM SETTING

For a given program C, a precondition Σ and a postcondition Φ, the correctness of the implementation can be checked by the correctness of the Hoare triple $\{\Sigma\}C\{\Phi\}$. However, if we want a specification that summarizes the bugs in the program C, it should be an underapproximate triple or we end up with false positive alerts.

The thing becomes more complicated because we do not simply want a proof of the underapproximate triple $[\Sigma]C[\neg\Phi]$ (most of the time this triple is wrong), we now need a new underapprximate triple $[P]C[Q]$ such that $P \implies \Sigma$ and $Q \implies \neg\Phi$. Furthermore, to avoid the trivial result that is $P = \Sigma$ and $Q = \bot$, we expect P and Q to be as precise (that is, the weakest Q and strongest P) as possible. Ideally, assuming that $R(\Sigma, C)$ indicates the set of reachable states from Σ by execution of C, we have the postcondition Q^* that is weakest and the precondition P^* that is strongest,

$$Q^* \equiv R(\Sigma, C) \wedge \neg\Phi$$
$$P^* \equiv R^{-1}(Q, C)$$

where R^{-1} is the inverse function of R.

Finding such a precise underapproximate triple $[P^*]C[Q^*]$ is hard (as the reachable function R is hard to find), so abstraction may be required. For concrete domain C (e.g. the set of program states), abstraction domain $D \subset C$ (e.g. logical formula built from method predicates), and corresponding mapping $\alpha : C \to D$ and $\gamma : D \to C$, we want to find a domain P^D just cover P^* (which is an strongest overapproximation) and a domain Q^D just covered by Q^* (which is an weakest underapproximation). Formally, we want to find a "precise" (precise with respect to the abstract domain) underapproximate triple $[P^D]C[Q^D]$, where the "precise" is defined as

(1) P^D is overapproximation: $P^* \implies \gamma(P^D)$;
(2) P^D is strongest: $\forall {P'}^D, (P^* \implies \gamma({P'}^D)) \implies (P^D \implies {P'}^D)$;
(3) Q^D is underapproximation: $\gamma(Q^D) \implies Q^*$;
(4) Q^D is weakest: $\forall {Q'}^D, (\gamma({Q'}^D) \implies Q^*) \implies ({Q'}^D \implies Q^D)$.

2 ALGORITHM

Even with the help of the abstract domain, the reachability analysis for the recursive program is challenging, as we need the help of inductive invariants. The typical inductive invariant inference tool like "IC3" based on the overapproximation does not here. For example, assume that an output set I' can be reached from input set I by the execution of transition T (e.g. body of a loop):

$$R(I, T) = I', I' \implies I$$

It suffices to say that I is an inductive invariant in the correctness logic. However, I is not inductive in the incorrectness logic, as it still covers some unreachable states (e.g. $I \wedge R(I, T)$). To get a real inductive invariant in incorrectness logic, we need a stronger assumption ([ZZ: not sure, if I am wrong please correct me]):

$$R(I, T) = I$$

2022. XXXX-XXXX/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

We should use a reverse IC3, which consists of a sequence of underapproximations from the output to the input.

In addition to these, the data can somehow accelerate the specification inference.

ACKNOWLEDGEMENTS

REFERENCES