Abstraction Domains

William Schultz

September 15, 2022

For the safety verification of transition systems, we typically must perform some kind of *abstraction.* For finite transition systems, verification is theoretically decidable, but practically it suffers from the state space explosion problem, and so exhaustive verification may be hard (e.g. exponential) in general. So, for any systems of non-trivial size, abstraction is typically necessary. Finding an inductive invariant to prove safety is, essentially, about finding a suitable abstraction that overapproximates the set of reachable system states. We presumably want this abstraction to be "concise" i.e. it is expressible in a form (exponentially?) more compact than the set of reachable states.

In general, in order to discover a "concise" inductive invariant we work over some *abstraction domain.* Given a state space S, we define an abstraction domain $D \subseteq 2^S$ as simply a set of subsets of S. For example, given the state space defined by a single real valued variable $x \in \mathbb{R}$, a possible abstraction domain is

$$D_1 = \{x > 2, x < -2\}$$

where each element of D_1 is a subset of $\mathbb{R}$, defined as a symbolic predicate over x.

One way to define an abstraction domain for a state space S is to explicitly define the set $D \subseteq 2^S$. Alternatively, we can provide a set of atomic predicates and rules for for how these predicates can be combined to form additional predicates. Our abstraction domain is then defined as the space of all possible composite predicates that can be formed as combinations of atomic predicates using these operators (perhaps up to some bounded size). We can consider this the *grammar-based* approach.

For example, for a state space S we can define a *grammar* G as a pair (P, O) where $P \subseteq 2^S$ is a set of predicates on S, and O is a set of operators for combining elements of P to form new predicates on S. These operators may be unary, binary, etc. For example, we may have a grammar $G_1 = (\{x > 2, x < 3\}, \{\neg, \vee\})$. The set O might be composed of symbolic/logical operators, but in general O may contain any set-based operators i.e. operators that take in some set of predicates of P and produce new predicates in 2^S.

For transition systems with a state space S, we can also always work over a "trivial" abstraction domain. That is, the domain $D_\bot = \{\{s\} \mid s \in S\}$ that consists of all "singleton" predicates i.e. those that contain a single concrete state. We can view this domain as "minimally abstract", since the predicates don't cover multiple states, and so don't really perform any "true" abstraction.