Abstraction for Model Checking

William Schultz

August 29, 2022

Abstraction, in the context of model checking, is generally aimed at reducing the size of the state space in an attempt to remove details that are irrelevant to the property being verified [1]. That is, broadly, abstraction is a fundamental tool in tackling the "state explosion" problem.

Abstraction for Kripke Structures

In general, an abstraction framework defines a set of concrete objects and abstract objects and a definition of how to map between them. For model checking, we typically use Kripke structures as our concrete objects. Recall that a *Kripe structure* $M = (AP, S, I, R, L)$ is defined as

- a set AP of atomic propositions
- a set of states S
- a set of initial states $I \subseteq S$
- a transition relation $R \subseteq S \times S$
- a labeling function $L : S \to 2^{AP}$

Simulation

To define a notion of abstraction for Kripke structures, we define a few standard relations between two structures M_1 and M_2. *Simulation* is a preorder (reflexive and transitive) in which the larger structure may have more behaviors, but possibly fewer states and transitions.

Let $M_1 = (AP_1, S_1, I_1, R_1, L_1)$ and $M_2 = (AP_2, S_2, I_2, R_2, L_2)$ be Kripke structures such that $AP_2 \subseteq AP_1$. A relation H is a *simulation relation from M_1 to M_2* if for every $s_1 \in S_1$ and $s_2 \in S_2$ such that $H(s_1, s_2)$, both of the following conditions hold:

- For all $p \in AP_2$, $s_1 \in L(s_1) \iff s_2 \in L(s_2)$
- $\forall t_1 : (R_1(s_1, t_1) \Rightarrow \exists R_2(s_2, t_2) \land H(t_1, t_2))$

We say that M_1 *is simulated by* M_2 (or M_2 *simulates* M_1) if there exists a simulation relation H from M_1 to M_2 such that

$$\forall s_1 \in I_1 : (\exists s_2 \in I_2 : H(s_1, s_2))$$

Bisimulation

Counterexample-Guided Abstraction Refinement (CEGAR)

If we start with some abstraction of our Kripke strucutre and try to model check it, we may encounter spurious errors. So, we use such a counterexample to refine our abstraction, and then repreat this process.

SAT-based Abstraction

See [2].

References

[1] Dennis Dams and Orna Grumberg. *Abstraction and Abstraction Refinement*, pages 385–419. Springer International Publishing, Cham, 2018.

[2] Kenneth L. McMillan and Nina Amla. Automatic abstraction without counterexamples. In *Proceedings of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'03, page 2–17, Berlin, Heidelberg, 2003. Springer-Verlag.