2

Verification problems

Systems are mathematical models of dynamical phenomena that allow for rigorous analysis. In this chapter we describe the two kinds of verification problems that are considered in this book.

2.1 $S_a \cong S_b$

The first verification problem is the *equivalence problem.*

Problem 2.1 (Equivalence). Given systems S_a and S_b and a notion of equivalence between systems, when is S_a *equivalent* to S_b?

If one denotes system equivalence by the symbol $\cong$, then Problem 2.1 asks when the following relationship holds:

$$S_a \cong S_b.$$

Several different analysis and verification problems arising in the design of complex systems can be casted as instances of the equivalence problem. This can be done for systems that have already been designed as well as for systems that have not yet, or have only been partially designed. In the former case, we regard S_a as a model of the system that has already been designed and S_b as a model of the specification. A positive answer to the equivalence problem would then imply that the design conforms to the specification. In the later case, we regard S_a and S_b as potential models of the same dynamical phenomenon and seek to determine if both models are equivalent. A positive answer to the equivalence problem would imply that any of the models could be used to complete the design at hand. In both cases we are implicitly assuming that one of the models is much simpler than the other. If S_b describes the specification then it is natural to expect that it should be much easier to construct S_b than S_a. When S_a and S_b are both models for the same system being designed, S_b being a much simpler model than S_a would guarantee that

P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*,
DOI: 10.1007/978-1-4419-0224-5_2, © Springer Science + Business Media, LLC 2009

the remaining design could be accomplished with greater ease by working with the simpler model S_b. This observation immediately places some restrictions on the notions of equivalence as they need to treat as equivalent, system S_a and the much simpler system S_b.

In this book we distinguish between two different kinds of equivalence: exact and approximate. While exact equivalence can be used for finite-state and infinite-state systems, approximate equivalence is more natural in the context of infinite-state systems describing dynamical, control, or hybrid systems. Exact equivalence requires the outputs of equivalent systems to be exactly the same while approximate equivalence relaxes this requirement by allowing the outputs to differ up to some specified precision. It is shown in Part IV that the additional flexibility afforded by approximate equivalence results in a larger class of infinite-state systems having equivalent finite-state symbolic models.

2.2 $S_a \preceq S_b$

In many circumstances the equivalence problem may be too demanding. If S_b is a model for the specification, it may be impossible to design a system S_a that is equivalent to S_b. However, S_a may still satisfy the specification in a weaker sense captured by the *pre-order problem*.

Problem 2.2 (Pre-order). Given systems S_a and S_b and a pre-order[1] between systems, when does S_a *precede* S_b?

If one denotes the pre-order by the symbol $\preceq$, then Problem 2.2 asks when the following relationship holds:

$$S_a \preceq S_b.$$

Intuitively, $S_a \preceq S_b$ is interpreted as S_a being "included" in S_b. The exact meaning of "included" will depend on the particular pre-order being used. As was the case with equivalence we will consider exact and approximate pre-orders, the later being a generalization of the former.

[1] Recall that a pre-order is a relation which is reflexive and transitive. See the Appendix for more details on pre-orders.

http://www.springer.com/978-1-4419-0223-8

Verification and Control of Hybrid Systems
A Symbolic Approach
Tabuada, P.
2009, XV, 202 p. 200 illus., Hardcover
ISBN: 978-1-4419-0223-8