

Methods for Mu-calculus Model Checking: A Tutorial

E. A. Emerson

Department of Computer Science, University of Texas at Austin, Austin, TX 78712
USA

The Mu-calculus may be viewed as a special type of temporal or modal logic. It permits characterization of many correctness properties of interest for parallel/reactive programs in terms of extremal fixpoints of predicate transformers. Mu-calculus formulas are built up from simple primitives: atomic propositions, boolean connectives, next-time operators, and least and greatest fixpoint operators. It turns out that, despite its simple underpinnings, the Mu-calculus is a very general and rich formalism. Most all correctness properties expressible in standard temporal logics, such as LTL, CTL, CTL*, are easily rendered in restricted fragments of the Mu-calculus.

Because of its simplicity and generality, there has been much interest in the potential of the Mu-calculus to serve as a sort of “universal” specification language. Some theoretical justification for its universality may be found in the recognition that the Mu-calculus coincides in expressive power with finite state automata on infinite trees. Moreover, many practical correctness properties have elegant Mu-calculus characterizations. Hence, there has also been considerable interest in developing efficient model checking algorithms for the Mu-calculus.

In this tutorial we survey various approaches to model checking in the Mu-calculus. The most basic idea is “fixpoint computation” using the Tarski-Knaster theorem to iteratively calculate a sequence of approximants converging on the fixpoint. This method underlies most “global” methods, whether implemented extensionally or symbolically with BDDs. Such global methods calculate the entire set of states where the formula holds. We will also discuss “local” methods which are intended to evaluate a specification at a specific state. In practice, local methods can be advantageous because they often may require exploration of only a limited portion of the global state space. We then discuss a notion of local model checking over infinite state spaces. We go on to describe how model checking in the Mu-calculus can be performed through testing nonemptiness of tree automata, and through solving boolean equation systems. Finally, we consider some theoretical complexity issues.