Branching temporal logics, automata and games

*B*ackground

The satisfiability problem for branching-time temporal logics like CTL and CTL+ has important applications in program specification and verification. Their computational complexities are known: CTL * and CTL+ are complete for doubly exponential time, CTL is complete for single exponential time. Some decision procedures for these logics are known; they use tree automata, tableaux or axiom systems. Automata-theoretic approaches. As much as the introduction of CTL * has led to an easy unification of CTL and LTL, it has also proved to be quite a difficulty in obtaining decision procedures for this logic. The first procedure by Emerson and Sistla was automata-theoretic [ES84] and roughly works as follows. A formula is translated into a doubly-exponentially large tree automaton whose states are Hintikka-like sets of sets of sub formulas of the input formula.

This tree automaton recognizes a superset of the set of tree models of the input formula. It is lacking a mechanism that ensures that certain temporal operators are really interpreted as least fix points of certain monotone functions rather than arbitrary fix points. Other approaches. Apart from these automata-theoretic approaches, a few deferent ones have been presented as well. For instance, there is Reynolds' proof system for validity [Rey01]. Its completeness proof is rather intricate and relies on the presence of a rule which violates the sub formula property. In essence, this rule quantity over an arbitrary set of atomic propositions. Thus, while it is possible to check a given tree for whether or not it is a proof for a given formula, it is not clear how this system could be used in order to find proofs for given formulas.

*S*ignificance

Advantages of the game-based approach. The game-theoretic framework uniformly treats the standard branching-time logics from the relatively simple CTL to the relatively complex. It yields complexity-theoretic optimal results, i.e. satisfiability checking using this framework is possible in exponential time for CTL and doubly exponential time for CTL+. Like the automata-theoretic approaches, it separates the characterization of satisfiability through a syntactic object (a parity game) from the test for satisfiability (the problem of solving the game). Thus, advances in the area of parity game solving carry over to satisfiability checking. Like the tableaux-based approach, it keeps a very close relationship between the input formula and the structure of the parity game thus enabling feedback from a (counter-)model or applications in specification and verification. Satisfiability checking procedures based on this framework are implemented in the MLSolver platform [FL10] which uses the high-performance parity game solver PG-Solver [FL09] as its algorithmic backbone.