TEMPORAL LOGIC MODEL CHECKERS AS APPLIED IN COMPUTER SCIENCE

Kazimierz Trzęsicki

Katedra Logiki, Informatyki i Filozofii Nauki Uniwersytet w Białymstoku

Various logics are applied to specification and verification of both hardware and software systems. Since systems are operating in time, temporal logic is a proper tool. The problem with finding of proof is the most important disadvantage of proof-theoretical method. Proprieties of a system can also be checked using a model of the system. The model is constructed with the specification language and checked using automatic model checkers. Model checking application presupposes the decidability of the task. The explosion of the cases that have to be explored is the main disadvantage of this method.

Temporal logic model checking is an algorithmic method that can be used to check whether a given model (representing a system) satisfies certain properties (expressed as temporal logic formulas).

In 1974 Burstall remarked the possibility of applications of modal logic to solve problems of computer science (*CS*). The Dynamic Logic of Programs has been invented by Pratt:

> In the spring of 1974 I was teaching a class on the semantics and axiomatics of programming languages. At the suggestion of one of the students, R. Moore, I considered applying modal logic to a formal treatment of a construct due to C. A. R. Hoare, "$p\{a\}q$", which expresses the notion that if p holds before executing program a, then q holds afterwards. Although I was skeptical at first, a weekend with Hughes and Cresswell convinced me that a most harmonious union between modal logic and programs was possible. The union promised to be of interest to computer scientists because of the power and mathematical elegance of the treatment. It also seemed likely to interest modal logicians because it made a well-motivated and potentially very fruitful connection between modal logic and Tarski's calculus of binary relations.

The question of using of temporal logic (TL) to software engineering was undertaken by Kröger. The development of TL as applied to CS is due to Pnueli. He was inspired by „Temporal Logic", a book written by Rescher and Urquhart. „The Temporal Logic of Programs", a work by Pnueli, is the classical source of TL for specification and verification of programs. This work is commonly seen as the turning point of using of TL in CS.

The reduction of errors in computer systems is one of the most important challenges of CS. Errors should already be detected at design stage. It is estimated that 70% of design-time is spent to minimize the risk of errors. Formal methods are proposed as efficient and less expensive tools. Temporal logic and its language are of particular interest in the case of concurrent and reactive systems. Today the knowledge of TL is indispensable in practice, tough, as it is remarked by Schnoebelen:

> In today's curricula, thousands of programmers first learn about temporal logic in a course on model checking!

TL language could be used to specification of widely spectrum of systems. Methods of TL could be applied to verification. In the case of operational systems TL is more useful than Floyd-Hoare logic

[1] The work was supported by the KBN grant 3 T11 F011 30

that is better in the case of “input-output” programs. There are two main methods: proof-theoretical and model-theoretical.

Proof-theoretical method was proposed by Pnueli and Manna. By the end of sixties of last century Floyd, Hoare and Naur proposed axiomatic proving sequential programs with respect to their specification. Verification is positive if the proposition expressing the desired property is proved. Finding a proof needs some experience and insight.

Model-theoretical method is based on construction of a model that has to accurately describe the behavior of the checked system. If logic is complete with respect to the model and is decidable, then in the case of any proposition the procedure is finite and if the proposition is not valid the construction results in a counterexample. The counterexample provides an information about an error (bug) in the system.

Automatic verification of concurrent programs based on model-theoretic approach was proposed by Emerson and Clarke, and independently by Queille and Sifakis. The idea was developed in works by Clarke, Emerson and Sistla.

Various model checkers are developed. They are applied to verification of large models, to real-time systems, probabilistic systems, etc. In this paper we survey and classify TL model checkers.