Verification Support by Model Checking

Masahiro Nakano (110093)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 14, 2003

Keywords: Formal Specification, Deductive Verification, Model Checking, Behavioural Specification, Translater.

1 Introduction

It is important to show the correctness of various systems like a hardware system and a software system. We are interested in a formal verification for designing and implementation of a system which satisfies important properties. There are mainly two approaches for the method of verifying specification.

Deductive Approach has sufficient description capabilities and verification capabilities. A Verification needs heuristics such as discovering of a lemma. Although its success means the properties are satisfied, its failure does not mean that the properties are never satisfied . Since it has problems in either specification, properties and it's method, we need to retry to verify after modifying problems. It is, however, very difficult to detect where the problems are within a whole verification.

There is a behavioral specification which is able to describe infinite state machine. On behavioral specification a state is concealed and cannot be described clearly. It is observed a part of information on the present by observation operation and is changed by action. The information is described as a term, and the transition relation is given by the change of observed terms between a current state and the state applied action. The initial value is given by a term as an observed value for initial state.

Model Checking Approach is the verifying method which limited to finite state machine. The verification is completely automatic. Its success means formation of property and its failure can give a counter-example as the reason. It becomes an important key for correcting specification and property.

SMV is one of the most popular Model Checker. A state in SMV is represented the tuple of the variable which takes a limited value and is carried out by updating all variables at once by next operation. The transition relation is given by the change of value of variables between before and after applied next operation. The initial state is given by defining init operation.

Both methods are radically different, and it is expected to be more useful verifying tool by integrating them. As first step of this integration, by our research, the integrated tool proves a specification with infinite state by deductive approach, and use model checker as a test generator on specification. Finding a counter-example by model checker automatically, we can modify wrong specification and property at an early stage.

2 Approach

In this section, we define the translator from behavioral specification to SMV specification. The inputs satisfie the following as constrains.

- A state equivalence is given by each observation value equivalence.
- The abstract data types which can be used are boolean and natural number exactly.

There are some information which cannot be observed on behavioral specification. By the first constraint we can always observe all information. It makes possible to define a state with the tuple of variables from observation operations because of no hidden information. The second constraint makes possible to realize the translator, because it is not easy to translate the term on arbitrary abstract data types. Even if it gives these restrictions, an extensive problem can be dealt with, and verification can be supported by model checking.

Translation processing is performed by defining a state, next operation, and init operation from behavioral specification. A state is given by defining state variables corresponding to the data type that mapped values returned by observation operation. In order to define next operation to each state variable, the equations are classified by observation first. The condition and value applied an equation are taken out from each equation, and they are listed every classification to make up next operation. The condition of transition is made from equation condition, kinds of action and so on. The init operation is defined by an equation for initial value if the equation exists, or is defined as the arbitrary values which can been taken by the state variable.

The more essential problem on translation is the difference of specifications to treat. behavioral specification treats infinite state machine, and SMV treats finite specification. So a transformation of data type cannot map to SMV data type and try to add any number of variables. In this research, we adopt the by restriction to finite states. The restriction is processing which takes out some state machine, and is able to apply to behavioral specification easily and immediately. Although there is abstraction as same technology, it is difficult to acquire the abstracting method. Since the omission in inspection is produced by restriction, it becomes impossible to show the correctness of specification.

3 Conclusion

For behavioral specification we suggested the ways to verify by CafeOBJ and to test by model checker SMV, and we implemented the translator C-TRAS to change behavioral specification into SMV specification.

Model Checking to behavioral specification is conducted as follow. At first model checker tests the SMV specification generated by C-TRAS. We consider the reason and try to modify the specification if a counter-example is found. Or there are two stories.

(1) The error is contained in the state omitted by restriction.

(2) There is no error on behavioral specification.
In this case it is necessary to verify specification by CafeOBJ in order to show no error. In this research it is proved that a found counter-example which obtained by restriction is also a counter-example in original behavioral specification.

It is actually experimented to the mutual exclusion systems based on load and store commands, test & set command, and Peterson's algorithms. We checked that each counter-example reported by model checker is the counter-example in original specification by running a proof score in CafeOBJ. In the proof by deductive approach, since these counter-examples were not able to be acquired automatically, we confirmed the validity of the model checker to behavioral specification.

Since the model checking to the specification applied restriction may not discover the error of the original specification, it cannot be called verification. It is one of the test techniques, and the translator can be caught with the test generator on specification. Our approach has the feature that exhaustive test for limited state space and infinite path test. For the usual test, the test case of an infinite path cannot be described, and the exhaustiveness is not guaranteed. Our approach is more powerful at these points compared with the usual test, and contributed also in the improvement in reliability of the test to specification.