| | |
|---|---|
| Title | Spinを用いたバイナリモデル検査 |
| Author(s) | 土肥，雅俊 |
| Citation | |
| Issue Date | 2008-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/4354 |
| Rights | |
| Description | Supervisor:青木利晃，情報科学研究科，修士 |

# Binary Model checking with SPIN

Masatoshi Doi (0610060)

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

February 7, 2008

**Keywords:** spin, binary model checking, model checking, verification,formal method.

## 1 Introduction

Recently, society keeps watch on formal method to guarantee software's dependability, but it is difficult to verify software's behavior, and it is necessary to convert from programing language to model description language. In addition to that there are problems which state explosion and description ability differences between programing languages and model description languages. Even if technical experts were rich in experience, they could not solve these problems simply. So, I propose that we use the binary model checking method for C programs with SPIN's EMBEDDED C CODE. SPIN adopts PROMELA(PROCESS METAMODEL LANGUAGE) , and also has the ability of embedding the C language into the verification code directly. By making good use of this ability, we have a method which use a suitable verification model but doesn't change the C program's structure.

## 2 What is the Binary Model Checking

Binary Model Checking is a new verification method which is defined in this research. This method compiles C programs, executes binary data and

starts to verify at the same time. SPIN does model checking considering memory space used by C binary as state space. That is why, I decided to call this method "binary model checking". In addition, it is possible to verify programs in a standard environment. Advantages of this method are solving problem between programming language and model description language, making verification model automatically and making use of the existing resources.

## 3 Verification by Binary Model Checking

To establish the binary model checking method as a good verification method, I need to make a premise which make general discussion possible. So, first of all, I will define the premise of this verification method. In this verification method, I give a set of input variables to C programs, check a set of output variables which are produced by C programs, and I guarantee a correctness of C programs.

Next, I verified the sort program using the binary model checking method. As the result, I could guarantee the sort program's correct behavior.

Verification of the sort program succeeded, but the sort program is a very small-scale program. Then I selected the ls program which displays file system information to verify. The ls program is a middle-scale program which includes major algorithms and structures. It was established during this experiment that new problems exist during the verification process. For example, results of verification are affected by the condition of the computer system, because the ls program calls system call functions. Accordingly, I discussed the configuration of the verification environment and considered the cases separately. Since, the differences became more clear between qualities which can be guaranteed and qualities which can not be guaranteed, the binary model checking method proved to be valid. In addition to this, there were problems which related to the method to assigning memory or a number of states. However, the binary model checking process was not difficult and I could easily find software bugs.

- if it is a simple C program, the binary model checking method can verify the C program directly.

- There are some problems with verification of middle-scale programs, but the binary model checking process is simple.
- It is possible to reduce the number of states, if I don't assign useless variables as state vector.
- There are problems which rerate to dealing with dynamic memory space.

## 4 Summary

In this paper, I proposed an original method which is called the binary model checking method, and found some problems. First of all, I defined the premise of the binary model checking method. Second, I made the binary model checking method clear. Third, I considered about the advantages of binary model checking. Finally, I applied the binary model checking method to actual programs. As a result of this experiment, I can verify C programs without changing the original program structure. The above-mentioned is the result which be related to establishing verification methods.