JAIST Repository
https://dspace.jaist.ac.jp/

Title	モデル検査における誤り原因の特定に関する研究
Author(s)	小川，直哉
Citation	
Issue Date	2014-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12057
Rights	
Description	Supervisor:青木利晃　准教授，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Abstract of the Master's thesis

1210016 Naoya Ogawa

February 2014

The purpose of this paper is to propose a method to identify the cause of the error of verification with Spin that is tool of a model checking.

Model checking is one of the techniques to improve the reliability of software. In order to perform model cheking, it is necessary to write "M" that is a model of the system and to give "P" that means properties of the system. M is what a behavior of the system is modeled by using a language that is used for writing models. P is what a properties of the system is expressed by using a temporal logic formulas for the model. And, we can get whether the model satisfies property or not. Number of state M is limited. Therefore, if we search all of states cyclopaedically, we can check whether the model satisfies property or not.

If you make mistake to make a model or to give a property, a problem that we can't find any bug in the system occur. It is useless to verify with wrong model or property. Therefore, I propose a method to determine the cause, which making a model or giving a property is, of the error of verification.

A distinctive feature of this paper is that there is a prospect of identifying the cause of the error of verification even if writing a model of the system is not enough.

In this paper, I propose a method to determine the cause of an error by comparing the counter example with sampling sets that is prepared in advance. If you perform a model cheching, a counter example is outputed in the case of what M that is a model of the system doesn't satisfy P that means properties of the system. In the case of a model checking by Spin, sequence of execution that leads error situation is outputed as a counter example. But, this M and P is written by human who use it for model checking. It can't be said that a system is wrong just because a counter example is outputed. Sometimes a counter example is outputed because of a human error.

For example, let's consider to perform a model checking for a time a time-difference system traffic signal which has four signals(signal$A$, signal$B$, signal$C$, and signal$D$) as a target. We check that color of the signal changes in order as a property. To express a model of signals that changes its color as they are synchronizing each other is one of the way to write a behavior of a time a time-difference system traffic signal. In any other way, there is a way to use a controller that send orders to each signals by message communication. A property is expressed by a temporal logic formulas. We suppose to use this temporal logic formulas : `"r → [](p U q)"`. `"r"` means that a color of a signal is blue. `"p"` means that a color of a signal is blue. `"q"` means that a color of a signal is yellow. Therefore, `"r → [](p U q)"` should means that

a color of a signal remains blue until yellow. After we perform a model checking with this formula, it is assumed that next sequence of execution is outputed as a counter example. `(blue, red, blue, red)` $\rightarrow$ `(yellow, red, blue, red)` $\rightarrow$ `(red, red, blue, red)`. The sequence of execution is expressed by a set of four variable that take blue or yellow or red.First variable is a color of signal$A$, second variable is a color of signal$B$, third variable is a color of signal$C$, and last variable is a color of signal$D$. But, the first variable changes in order. This sequence of execution taht satisfy the property is a false counterexample. That a false counterexample is outputed as a counter example means that M has no problem.

Therefore, we can identify that there is an error in the P. In this way, in order to identify where an error is, it is necessry to prepare basis for juging whether a counter example is a false counterexample. In this paper, I will prepare "positive sampling sets" and "negative sampling sets". Positive sampling sets is things that we are confident that it is befavior that satisfies properties of the system. Negative sampling sets is things that we are confident that it is not behavior of the system. Using the example of a traffic signal, `(blue, red, blue, red)` $\rightarrow$ `(yellow, red, blue, red)` $\rightarrow$ `(red, red, blue, red)` is one of positive sampling sets. `(blue, red, blue, red)` $\rightarrow$ `(yellow, red, blue, red)` $\rightarrow$ `(blue, red, blue, red)` is one of negative sampling sets.

The cause of an error is determined by comparing the counter example with positive sampling sets or negative sampling sets that is prepared in advance. First of all, in order to compare with a counter example,it is necessary to decide how to write sampling sets.

As a result of experiments, I could find an effective method of writing negative sampling sets and propose a method to decide whether an error is contained in negative sampling sets. But, concerning positive sampling sets, I could't find an effective method of writing and method to decide whether an error is contained in positive sampling sets.

About method of writing negative sampling sets, I decided to use a transition of variables and conditional expression for variables. If you write odd expression in conditional expression, the transition of variables is a part of negative sampling sets.

Using the example of a traffic signal, $''u_1 \equiv (A, B, C, D) \Rightarrow (A', B', C', D')\ when\ (A = blue \wedge A' = red) \vee (A = yellow \wedge A' = blue) \vee (A = red \wedge A' = yellow)''$ is one of example. "$(A = blue \wedge A' = red)$" means that signalA turn to red from blue. "$(A = yellow \wedge A' = blue)$" means that signalA turn to blue from yellow. "$(A = red \wedge A' = yellow)$" means that signalA turn to yellow from red. Therefore, this conditional expression is odd expression. If some sequence of transition variables contains odd transition of variable, sequence of transition variables is a element of negative sampling sets. And, I created a pseudo-code that express a mthod to determine the mistake of making a model.

If I don't find an effective method of writing sampling sets, I can not propose a method to determine the cause of the error. Therefore, concerning consideration on positive sampling sets, I remark on a method of writing positive sampling sets.