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

Title	[課題研究報告書] An investigation of a state machine visualization tool SMGA & case studies with SMGA
Author(s)	小林, 翠
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17096
Rights	
Description	Supervisor: 緒方 和博, 先端科学技術研究科, 修士(情報科学)

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

An investigation of a state machine visualization tool SMGA & case studies with SMGA

1910093 Midori Kobayashi

Nowadays, the Internet and other software systems are used everywhere. For example, e-commerce systems such as Amazon and Rakuten have permeated our lives and are an integral part of our daily lives. If these systems do not work as intended, they may cause economic loss or human damages, which will not make our lives more convenient, but rather cause serious consequence. And the most core softwares of the Internet is a distributed system. It is not easy to develop a distributed system as desire because the systems are often concurrent programs. For example, in concurrent program, many processes and computers must use shared resources such as memory to meet certain constraints, so it is known that it is very difficult to develop a distributed system to work as intended and does not behave in any other way. To develop a distributed system (worked as intended), many technologies need to be used strictly. One such technique is formal verification. Formal verification can be categorized into model checking and theorem proving. Model checking often is used to detect defects, and theorem proving is necessary to ensure that the system works as intended. However, theorem proving often needs lemmas, which require much human effort. Finding lemmas is a big barrier of theorem proving in formal verification. On the other hand, humans are good at visual perception. State Machine Graphical Animation (SMGA) can be known as a tool that supports the use of theorem proving and information visualization. The tool has potential to remove or subordinate the barriers to lemma discovery.

The purpose of this study mainly is to survey case studies of SMGA. Maude is used to generate the state sequence of the state machine, which is the input of SMGA. For this reason, this research includes a survey on state machines, how to formalize protocols as state machines, how to describe state machines in Maude (how to create formal specifications), how to model check (the specifications) in Maude that state machines satisfy desired properties, and how to generate state sequences of state machines using Maude.

In this research report, five protocols will be studied. These include the Test & Set protocol (TAS), a flawed version of the Test & Set protocol (FTAS), the Qlock protocol, two flawed versions of the Qlock protocol (FQlock0 and FQlock1), and the Anderson protocol. Specifically, we will learn what kind of pseudocode these protocols are written in and what kind of rewriting rules are used. Then, we will explain how to visualize these protocols in SMGA and how to design them. Various figures will be used for the design. In this

way, we will not only visualize the design, but also make it easy to discover the characteristics. Then, we will describe what characteristics we found for each protocol by using SMGA. Finally, the correctness of these characteristics will be discovered by model checking. To create a graphical animation using SMGA, an input file is required. To create the input file, we need to generate the state sequence using Maude. Also, Maude is used for model checking. In Maude, we use the search command, one of the important commands, which can search for a user-specified state and confirm whether the state exists. Moreover, we will describe how to create a good diagram for graphical animation, how to observe graphical animations and look for protocol characteristics. These contents should be described concerning what characteristics you found in the animations you created in SMGA. We will then use these to help us discover even better ways to create diagrams and find characteristics of protocols.

One of the future tasks of the project report is to provide an important theorem proving in formal verification. We can perform model checking and formal verification to develop distributed systems as intended. Learning both of two techniques will be of great help in developing software that supports our daily lives. In addition to learning theorem proving, it is also important to learn how the applied protocols in the report are used in the software that we use in our daily lives. It will help us to further understand the report and learn about software development. In addition to the search command, there are many other commands in Maude. By using such that commands, we can perform various types of model checking. Model checking is excellent for finding defects in software development. In other words, using various commands is useful for finding various kinds of defects.