國立臺灣大學電機資訊學院電子工程學研究所

博士論文

Graduate Institute of Electronics Engineering

College of Electronical Engineering and Computer Science

National Taiwan University

Doctor Thesis

賽局上的合作時序邏輯的模型驗證以及密集錯誤回復力

Model Checking for Temporal Cooperation Logic and Resilience against Dense Errors on Game Graph

黃重豪

Chung-Hao Huang

指導教授：王凡博士

Advisor: Farn Wang, Ph.D.

中華民國 105 年 1 月

January, 2016

國立臺灣大學
電子工程學研究
所

博士論文

賽局上的合作時序邏輯的模型驗證以及密集錯
誤回復力

黃重豪

撰

105
1

Abstract

This thesis is composed of 3 parts. In the first part, I introduce an algorithm to calculate the highest degree of fault tolerance a system can achieve with the control of a safety critical systems. Which can be reduced to solving a game between a malicious environment and a controller. During the game play, the environment tries to break the system through injecting failures while the controller tries to keep the system safe by making correct decisions. I found a new control objective which offers a better balance between complexity and precision for such systems: we seek systems that are k-resilient. A system is k-resilient means it is able to rapidly recover from a sequence of small number, up to k, of local faults infinitely many times if the blocks of up to k faults are separated by short recovery periods in which no fault occurs. k-resilience is a simple abstraction from the precise distribution of local faults, but I believe it is much more refined than the traditional objective to maximize the number of local faults. I will provide detail argument of why this is the right level of abstraction for safety critical systems when local faults are few and far between. I have proved, with respect to resilience, the computational complexity of constructing optimal control is low. And a demonstration of the feasibility through an implementation and experimental results will be in following chapters. The second part is to create an logic which can describe the different purposes of each player such as environment, controller, user, and etc in a system. I propose an extension to ATL (alternating-time logic), called BSIL(basic strategy-interaction logic), for the specification of strategies interaction of players in a system. BSIL is able to describe one system strategy that can cooperate with several strategies of the environment for different requirements. Such properties are important in practice and I show that such properties are not expressible in ATL*, GL (game logic), and AMC (alternating μ-calculus). Specifically, BSIL is more expressive than ATL but incomparable with ATL*, GL, and AMC in expressiveness. I show that, for fulfilling a specification in BSIL, a memoryful strategy is necessary. I also show that the model-checking complexity of BSIL is PSPACE-complete and is of lower complexity than those of ATL*, GL, AMC, and the general strategy logics. Which may imply that BSIL can be useful in closing the gap between large scale real-world projects and the time consuming game-theoretical results. I then show the feasibility of our techniques by implementation and experiment with our PSPACE model-checking algorithm for BSIL. The final part is an extension to BSIL called temporal cooperation logic(TCL). TCL allows successive definition of strategies for agents and agencies. Like BSIL the expressiveness of TCL is still incomparable with ATL*, GL and AMC. However, it can describe deterministic Nash equilibria while BSIL cannot. I prove that the model checking complexity of TCL is EXPTIME-complete. TCL enjoys this relatively cheap complexity by disallowing a too close entanglement between cooperation and competition while allowing such entanglement leads to a non-elementary complexity. I have implemented a model-checker for TCL and shown the feasibility of model checking in the experiment on some benchmarks.

Key words:

Contents

Chapter 1

Introduction

The organization of this thesis is as follows. In chapte **??**, the theoretical background and definition of surface plasmon will be included [**?**]. Chapte **??** contains description of experiment methods such as atomic force microscopy and scanning electron microscopy.

Chapter 2

Software Resilience against Dense Errors

2.1 Two-player concurrent game structures

2.2 Motivation

2.2.1 Background

2.2.2 Resilience in a Nutshell

2.3 Safety resilience games

2.4 Alternating-time μ-calculus with events

2.4.1 Syntax

2.4.2 semantics

2.5 Resilience level checking algorithm

2.5.1 High-level description of the algorithm

2.5.2 Realization with AMCE model-checking

[illegible]

Chapter 3

Basic Strategy-interaction Logic

3.1 Existed Logics about Game and Strategy

3.1.1 Prior to Strategy Logics

3.1.2 With Strategy Logics

3.2 Running Example

3.2.1 Trying to Write Down a Correct Formal Specification

3.2.2 Resorting to General Strategy Logics for a Correct Specification

3.2.3 BSIL: New Strategy Modalities Expressive Enough for the Specification

3.2.4 Symbolic Strategy Names and Path Obligations

3.2.5 Passing Down the Path Obligations While Observing the Restrictions Among S-Profiles

3.2.6 Finding Finite Satisfying Evidence for a Formula in a Computation Tree

Chapter 4

Temporal Cooperation Logic

4.1 TCL

4.1.1 Syntax

4.1.2 semantics

4.2 Expressive Power of TCL

4.2.1 Comparison with CTL* and LTL

4.2.2 Comparison with AMC, ATL* and GL

4.3 Complexity

4.3.1 TCL Model-checking

4.3.2 TCL Satisfiability

4.4 Experiment

Chapter 5

Conclusions