COLLEGE OF COMPUTING AND INFORMATION SCIENCES DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

April 19, 2017

NAME: OKIRING PAUL
STUDENTS NUMBER: 214016765
REGISTRATION NUMBER: 14/U/13973/EVE

1 INTRODUCTION

Model checking has emerged as a powerful method for the formal verification of programs. Temporal logics such as CTL (computational tree logic) and CTL* are widely used to specify programs because they are expressive and easy to understand. Given an abstract model of a program, a model checker (which typically implements the acceptance problem for a class of automata) verifies whether the model meets a given specification. A conceptually attractive method for solving the model checking problem is by reducing it to the solution of (a suitable subclass of) parity games. These are a type of two player infinite game played on a finite graph.

2 PROBLEM STATEMENT

Given a model of a system, exhaustively and automatically check whether this model meets a given specification. Typically, one has hardware or software systems in mind, whereas the specification contains safety requirements such as the absence of deadlocks and similar critical states that can cause the system to crash. Model checking is a technique for automatically verifying correctness properties of finite-state systems. In order to solve such a problem algorithmically, both the model of the system and the specification are formulated in some precise mathematical language. To this end, the problem is formulated as a task in logic, namely to check whether a given structure satisfies a given logical formula. This general concept applies to many kinds of logics and suitable structures. A simple model checking problem is verifying whether a given formula in the propositional logic is satisfied by a given structure.

3 BACKGROUND

Model checking did not arise in a historical vacuum. There was an important problem that needed to be solved, namely concurrent problem verification. Concurrency errors are particularly difficult to find by program testing, since they are often hard to reproduce. Most of the formal research on this topic involved constructing proofs by hand using Floyd-Hoare style logic. Probably, the best known formal system was the one proposed by Owicki and Gries for reasoning about conditional critical regions. Also, in the late 1970s, Pnueli and Owicki and Lambert had proposed the use of temporal logic for specifying concurrent programs. Although they still advocated hand constructed proofs, their work demonstrated convincingly that temporal logic was ideal for expressing concepts mutual exclusion, absence of deadlock, and absence of starvation.

4 MAIN OBJECTIVES

To show the connexions between the temporal logics CTL and / or CTL*, automata, and games.

4.1 OBJECTIVES

1. Representing CTL / CTL* as classes of alternating tree automata

2. Inter-translation between CTL / CTL* and classes of alternating tree automata

3. Using Buchi games and other subclasses of parity games to analyze the CTL / CTL* model checking problem

4. Efficient implementation of model checking algorithms

5. Application of the model checker to higher-order model checking