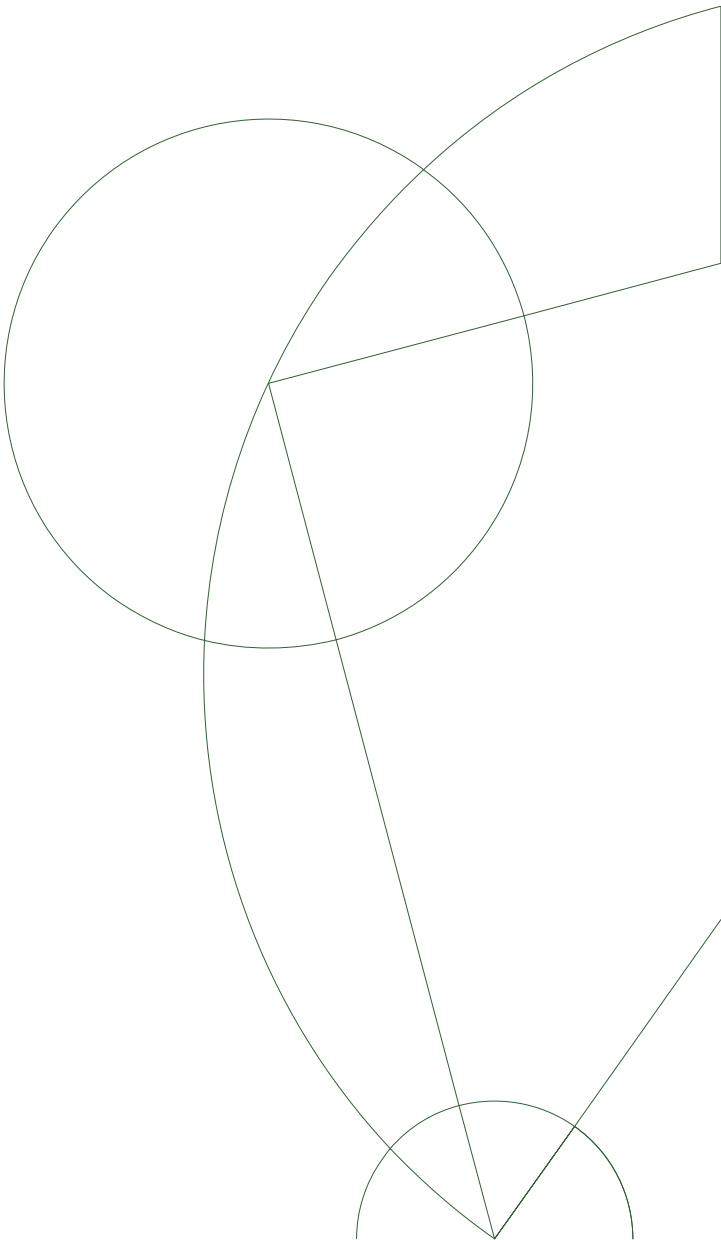
FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

Master's Thesis

Alberte Thegler - alberte@thegler.dk

Towards formal verification of FDR4

Department of Computer Science



Professor Brian Vinter

August 2018

Abstract

Bla bla bla bla

Contents

Todo list

1 Introduction

When we create programs, we wish to verify that it is also correct. There are several ways to do this, one commenly used is `testing` which require that the programmer creates several different scenarios and its expected output, or that the programmer programs a test-generator to create the scenarios and expected output. This, however, is not adequate for (word for important systens). Therefore it is of high interest to create a verification of the system or program.
Talk about how verification was first created and how it became to be used for concurrent systems. Then write about how it works and then write about the different systems and formal languages that is used for it.

In this thesis we look at model checking, that is, verifying that a specific property will always hold for a piece of code.

2 Related work

Related work should be about describing the history of formal verification and formal languages in both software and hardware. Formal verification is the process of checking whether a program satisfies specific properties. Different methods have evolved, all having different advantages and disadvantages. FDR is sometimes referred to as a model checker however is it actually a refinement checker.

In 1969, Tony Hoare proposed the Hoare logic[2] which is a formal system that can reason about the correctness of computer programs by using a set of logical rules. A lot of verification systems used today is based on Hoare logic. Hoare was much inpired by Robert W. Floyd, who a few years before had published his paper *Assigning meaning to programs*[1] where he created a system for flowcharts that had similar functionality.

2.1 Software Verification

Several different software verification tools exists today, and they all have different advantages.

SPIN

SPIN is a verification system that uses process interactions to prove correctness for a system. The system is described in the formal language `PROMELA`(PROcess MEta LAnguage)and the correctness properties are spcified in Linear Temporal Logic or LTL . Spin was developed at Bell Labs, starting in 1980. Since 1991 it has been freely available and today it is used by thousands of people worldwide. The job is to find other types of models and compare them, as well as the types of languages. so different tools and different languages..

Add reference to SPIN

add reference to promela

add reference to Linear Temporal Logic

Occam-pi is a programming language that came from occam and was merged with pi-calculus.

2.2 Hardware Verification

The job here is to find some examples of stuff that is relevant to SME and then see how I can compare them.

References

[1] R. W. Floyd. Assigning Meanings to Programs. pages 19–32, 1967.

[2] C. A. R. Hoare. An axiomatic basis for computer programming, 1969.