

Addressing the Performance Bottlenecks of Device Driver Synthesis

Alexander Legg

Supervised by
Leonid Ryzhyk, Nina Narodytska, and Gernot Heiser

August 13, 2015

Contents

1	Introduction	1
1.1	Device Drivers	1
1.2	Synthesis	1
2	Background	2

Abstract

Device drivers are notorious for their contribution to the failure of software systems. A driver forms the interface between two complicated systems: an operating system and a hardware device. As a result, driver development is complicated and highly prone to human error. Driver reliability has been the focus of much research over the past two decades. One result of this research is device driver synthesis.

Device driver synthesis aims to remove human error from the development process by removing humans. A synthesis tool can take a formal specification of requirements and produce a system. Thus, a driver can be synthesised by providing the tool with a specification of the OS and the hardware that it must interface with. The resulting driver is guaranteed to be correct with respect to the specifications.

The drawback of this approach is that synthesis is a computationally intensive task belonging to 2-EXPTIME. In practice synthesis tools are capable of producing drivers though there remain cases where synthesis is impossible. In this thesis I will present an algorithm that provides results on many of these previously unattainable cases.

Acknowledgements

Acknowledge some people

Chapter 1

Introduction

This is the introduction.

1.1 Device Drivers

1.2 Synthesis

Chapter 2

Background

This is the background.