The University of York Department of Computer Science

Submitted in part fulfilment for the degree of MEng.

Tracing and Debugging in GP2

Joshua Asch

Date TBC

Supervisor: Detlef Plump

Number of words = 0, as counted by `wc -w`.
This includes the body of the report only.

Abstract

This is my project!

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introductiony Bit

A section introducing the project.

1.2 Ethics

A section discussing the ethics of the project.

2 Literature Review

2.1 Programming by Graph Transformation

2.1.1 Graph Transformations

A section discussing what a graph transformation is, including rules, matches, interface nodes, etc.

2.1.2 The GP2 Language

Programs in GP2

A section discussing programs in GP2, including definitions and an overview of the grammar (possibly?).

The GP2 Compiler

A section about how a compiled GP2 program works. Section may be unnecessary?

The GP2 IDE

A section describing the IDE and how it relates to the compiled program.

Would it be appropriate to discuss what we want to do to the editor here?

Section may be unnecessary?

2.2 Program Tracing

2.2.1 Tracing in Imperative Languages

A section discussing what tracing and debugging imperative languages like C and Java is like.

2.2.2 Tracing in Functional Languages

A section discussing what tracing functional languages like Haskell is like. A main focus on the Hat tool for Haskell, describing how it relates to the problem of tracing GP2.

3 Another chapter...