

Contents

| | |
|--|-----------|
| Abstract | 2 |
| Acknowledgements | 3 |
| 1 Introduction | 7 |
| 2 Background | 8 |
| 2.1 Answer Set Programming | 8 |
| 2.1.1 The Stable Model Semantics | 8 |
| 2.1.2 Calculating Answer Sets | 9 |
| 2.1.3 Learning from Answer Sets | 11 |
| 2.1.4 ASP Solvers | 13 |
| 2.2 Inductive Functional Programming | 13 |
| 2.2.1 Conditional Constructor Systems | 13 |
| 2.2.2 Overview of current tools | 14 |
| 2.3 The Target Language, Haskell | 15 |
| 3 The Initial Approach : A Haskell Interpreter in ASP | 16 |
| 3.1 A Haskell Interpreter in ASP | 16 |
| 3.1.1 Target Language | 16 |
| 3.1.2 Program Representation | 16 |
| 3.1.3 Evaluating Rules | 17 |
| 3.2 Initial Learning | 21 |
| 3.3 Performance Issues | 22 |
| 4 A Second Approach : Constraint Based Learning | 23 |
| 4.1 Top Down Vs. Bottom Up | 23 |
| 4.2 Learning | 23 |
| 4.3 Performance | 23 |
| 5 Front end implementation : Building a working UI | 24 |
| 5.1 User's Manual | 24 |
| 5.2 Used Technologies | 24 |
| 5.2.1 Play Framework | 24 |
| 5.3 User Feedback and Evaluation | 24 |
| 6 Critical Evaluation | 25 |
| 6.1 Testing | 25 |
| 6.1.1 One Argument Programs | 25 |

| | | |
|----------|--|-----------|
| 6.1.2 | Two Argument Programs | 25 |
| 6.2 | Comparison to Existing Tools | 25 |
| 7 | Conclusions and Future Work | 26 |
| 7.1 | Conclusions | 26 |
| 7.1.1 | What worked | 26 |
| 7.1.2 | Areas for improvement | 26 |
| 7.2 | Future Work | 26 |
| 7.2.1 | Learning Improvements | 26 |
| 7.2.2 | UI Features | 26 |