Contents

Introduction 1

Contributions of the Thesis . . . 3

Part I – Backward Trace Analysis . . . 3

Part II – Forward Trace Analysis . . . 5

Related Work . . . 7

0 Preliminaries **11**

0.1 The Term-language of Maude . . . 11

0.2 Program Equations and Rules . . . 13

0.3 Conditional Rewrite Theories . . . 14

0.4 Rewriting in Conditional Rewrite Theories . . . 18

0.5 Instrumented Execution Traces . . . 21

0.6 Term Slices and their Concretizations . . . 25

0.7 (Instrumented) Trace Slices and their Concretizations . . . 27

0.8 Meaningful Descendants and Ascendants . . . 28

I Backward Trace Analysis **31**

1 Backward Trace Slicing for Conditional Rewrite Theories **33**

1.1 Backward Slicing for Execution Traces . . . 34

1.2 The Function *slice-step* . . . 39

1.3 Correctness of Backward Trace Slicing . . . 45

2 The *i*Julienne System **51**

2.1 *i*Julienne at Work . . . 53

2.1.1 Debugging Maude Programs with *i*Julienne . . . 53

2.1.2 Trace Querying with *i*Julienne . . . 61

2.1.3 Dynamic Program Slicing . . . 68

2.2 Experimental Evaluation . . . 70

II Forward Trace Analysis 73

3 Exploring Conditional Rewriting Logic Computations 75
3.1 The Generic Exploration Scheme 76
3.1.1 Inspecting the Instrumented Traces 77
3.1.2 Exploring the Instrumented Computation Tree Slices 79

4 Exploration Modalities 83
4.1 Interactive Stepper 83
4.2 Partial Stepper 84
4.3 Stepper and Partial Stepper Correctness 87
4.4 Forward Trace Slicer 90
4.5 Forward Trace Slicer Correctness 98
4.6 Backward Trace Slicing as an Instance of the Generic Scheme 101

5 The Anima system 105
5.1 The Anima Exploration Tool 105
5.2 Implementation of the Tool 111

Conclusions 117

Bibliography 121

A Maude Specification of the Experimental Evaluation Examples 133

List of Figures

1 Maude specification of the maze game. 16

2 The 5×5 grid encoded in `MAZE` 17

3 Meaningful descendants of a rewrite step $s \xrightarrow{r,\sigma,w}_K t$. 28

1.1 The `_mod_` operator. 34

1.2 Maude specification of a distributed banking system. . . 38

1.3 Backward step slicing function. 39

1.4 Condition processing function. 41

2.1 *i*JULIENNE architecture. 52

2.2 Faulty Maude specification of a distributed banking system. 54

2.3 *i*JULIENNE output for the trace $\mathcal{T}_{bank}$ w.r.t. the slicing criterion `ac( • ,-11)`. 55

2.4 `BLOCKS-WORLD` faulty Maude specification. 57

2.5 Program slice computed w.r.t. the slicing criterion `empty & empty`. 58

2.6 Navigation through the trace slice of the *Blocks World* example. 60

2.7 Navigation though the refined trace slice of the *Blocks World* example. 60

2.8 Program slice computed w.r.t. the slicing criterion `hold(•)`. 61

2.9 Maude specification of the `minmax` function. 63

2.10 *i*JULIENNE output for the trace $\mathcal{T}^{\bullet}_{minmax}$ w.r.t. the slicing criterion automatically inferred in the query `PAIR(?,_)`. . 64

2.11 Loading the webmail execution trace. 65

2.12 Querying the webmail trace w.r.t. `B(idA,_,?,_,_,_,_,_,_)`. 66

2.13 Webmail trace slice after querying the trace. 67

2.14 `MINMAX` program slice computed w.r.t. the query `PAIR(?,_)`. 69

3.1 Computation tree. 76

3.2 The inference rule trans of the transition system $(Conf, \Longrightarrow)$. 78

3.3 The one-step *expand* function. 80

3.4 The interactive *explore* function. 81

4.1 Inspection of the state s_0 w.r.t. $\mathcal{I}_{step}$. 84
4.2 Computation tree slice fragment for $s_0^{\bullet}$ w.r.t. $\mathcal{I}_{pstep}$. . . . 86
4.3 Inspection function that models the forward slicing of a conditional rewrite step. 91
4.4 The condition processing function. 92
4.5 Computation tree slice fragment for $s_0^{\bullet}$ w.r.t. $\mathcal{I}_{slice}$. . . . 97

5.1 Anima architecture. 106
5.2 Anima at work. 107
5.3 Anima search mechanism. 108
5.4 Anima trace information. 109
5.5 Inspection of a condition with Anima. 111
5.6 Benchmark problem for the `metaReducePath` command. 116