Contents

Preface by Luca Cardelli VII

Prologue XV

Reading Path XIX

List of Figures XX

List of Tables XXIV

Definitions and Properties XXVII

Part I Review 1

1 Analysis 3

1.1 A Few Definitions . . . 3

1.2 Distribution, Parallelism, Concurrency . . . 5

1.2.1 Parallel Activities . . . 5

1.2.2 Sharing . . . 6

1.2.3 Communication . . . 6

1.2.4 Synchronization . . . 10

1.2.5 Reactive vs. Proactive vs. Synchronous . . . 11

1.3 Objects . . . 14

1.3.1 Object vs. Remote Reference and Communication . . . 14

1.3.2 Object vs. Parallel Activity . . . 14

1.3.3 Object vs. Synchronization . . . 15

1.4 Summary and Orientation . . . 17

2 Formalisms and Distributed Calculi 21

2.1 Basic Formalisms . . . 21

2.1.1 Functional Programming and Parallel Evaluation . . . 21

2.1.2 Actors . . . 23

2.1.3 π-calculus . . . 26

2.1.4 Process Networks . . . 30

2.1.5 ς-calculus 31
2.2 Concurrent Calculi and Languages 35
2.2.1 MultiLisp 35
2.2.2 PICT 37
2.2.3 Ambient Calculus 40
2.2.4 Join-calculus 42
2.2.5 Other Expressions of Concurrency 43
2.3 Concurrent Object Calculi and Languages 45
2.3.1 ABCL 45
2.3.2 Obliq and Øjeblik 49
2.3.3 The $\pi o \beta \lambda$ Language 51
2.3.4 Gordon and Hankin Concurrent Calculus: **conc**ς-calculus 54
2.4 Synthesis and Classification 56

Part II ASP Calculus 59

3 An Imperative Sequential Calculus 63
3.1 Syntax 63
3.2 Semantic Structures 65
3.2.1 Substitution 65
3.2.2 Store 66
3.2.3 Configuration 66
3.3 Reduction 66
3.4 Properties 68

4 Asynchronous Sequential Processes 69
4.1 Principles 69
4.2 New Syntax 71
4.3 Informal Semantics 71
4.3.1 Activities 72
4.3.2 Requests 73
4.3.3 Futures 73
4.3.4 Serving Requests 73

5 A Few Examples 75
5.1 Binary Tree 76
5.2 Distributed Sieve of Eratosthenes 77
5.3 From Process Network to ASP 79
5.4 Example: Fibonacci Numbers 80
5.5 A Bank Account Server 81

Part III Semantics and Properties 83

6 Parallel Semantics 87

6.1 Structure of Parallel Activities 87
6.2 Parallel Reduction 89
6.2.1 More Operations on Store 89
6.2.2 Reduction Rules 91
6.3 Well-formedness 98

7 Basic ASP Properties 101
7.1 Notation and Hypothesis 101
7.2 Object Sharing 104
7.3 Isolation of Futures and Parameters 105

8 Confluence Property 107
8.1 Configuration Compatibility 107
8.2 Equivalence Modulo Future Updates 111
8.2.1 Principles 113
8.2.2 Alias Condition 114
8.2.3 Sufficient Conditions 115
8.3 Properties of Equivalence Modulo Future Updates 117
8.4 Confluence 118

9 Determinacy 121
9.1 Deterministic Object Networks 121
9.2 Toward a Static Approximation of DON Terms 124
9.3 Tree Topology Determinism 126
9.4 Deterministic Examples 126
9.4.1 The Binary Tree 126
9.4.2 The Fibonacci Number Example 127
9.5 Discussion: Comparing Request Service Strategies 130

Part IV A Few More Features 133

10 More Confluent Features 137
10.1 Delegation 137
10.2 Explicit Wait 141
10.3 Method Update 141

11 Non-Confluent Features 143
11.1 Testing Future Reception 143
11.2 Non-blocking Services 144
11.3 Testing Request Reception 145
11.4 Join Patterns 146
11.4.1 Translating Join Calculus Programs 146
11.4.2 Extended Join Services in ASP 147

12 Migration 151

12.1 Migrating Active Objects 151
12.2 Optimizing Future Updates 153
12.3 Migration and Confluence 154

13 Groups 157
13.1 Groups in an Object Calculus 157
13.2 Groups of Active Objects 160
13.3 Groups, Determinism, and Atomicity 162

14 Components 169
14.1 From Objects to Components 169
14.2 Hierarchical Components 170
14.3 Semantics 172
14.4 Deterministic Components 174
14.5 Components and Groups: Parallel Components 175
14.6 Components and Futures 177

15 Channels and Reconfigurations 179
15.1 Genuine ASP Channels 179
15.2 Process Network Channels in ASP 181
15.3 Internal Reconfiguration 182
15.4 Event-Based Reconfiguration 183

Part V Implementation Strategies 185

16 A Java API for ASP: ProActive 187
16.1 Design and API 187
16.1.1 Basic API and ASP Equivalence 188
16.1.2 Mapping Active Objects to JVMs: Nodes 189
16.1.3 Basic Patterns for Using Active Objects 190
16.1.4 Migration 190
16.1.5 Group Communications 193
16.2 Examples 196
16.2.1 Parallel Binary Tree 196
16.2.2 Eratosthenes 199
16.2.3 Fibonacci 204

17 Future Update 211
17.1 Future Forwarding 211
17.2 Update Strategies 213
17.2.1 ASP and Generalization: Encompassing All Strategies . 213
17.2.2 No Partial Replies and Requests 215
17.2.3 Forward-Based 217
17.2.4 Message-Based 218
17.2.5 Lazy Future Update 220

17.3 Synthesis and Comparison of the Strategies 221

18 Loosing Rendezvous 223
18.1 Objectives and Principles 223
18.2 Asynchronous Without Guarantee 225
18.3 Asynchronous Point-to-Point FIFO Ordering 227
18.4 Asynchronous One-to-All FIFO Ordering 230
18.5 Conclusion 233

19 Controlling Pipelining 235
19.1 Unrestricted Parallelism 236
19.2 Pure Demand Driven 236
19.3 Controlled Pipelining 237

20 Garbage Collection 239
20.1 Local Garbage Collection 239
20.2 Futures 240
20.3 Active Objects 240

Part VI Final Words 241

21 ASP Versus Other Concurrent Calculi 243
21.1 Basic Formalisms 243
21.1.1 Actors 243
21.1.2 π-calculus and Related Calculi 244
21.1.3 Process Networks 246
21.1.4 ς-calculus 247
21.2 Concurrent Calculi and Languages 247
21.2.1 MultiLisp 247
21.2.2 Ambient Calculus 248
21.2.3 join-calculus 248
21.3 Concurrent Object Calculi and Languages 248
21.3.1 Obliq and Øjeblik 248
21.3.2 The $\pi o\beta\lambda$ Language 249

22 Conclusion 251
22.1 Summary 251
22.2 A Dynamic Property for Determinism 252
22.3 ASP in Practice 253
22.4 Stateful Active Objects vs. Immutable Futures 254
22.5 Perspectives 255

23 Epilogue 259

Appendices 264

A Equivalence Modulo Future Updates 267

A.1 Renaming . . . 267
A.2 Reordering Requests ($R_1 \equiv_R R_2$) . . . 267
A.3 Future Updates . . . 268
A.3.1 Following References and Sub-terms . . . 268
A.3.2 Equivalence Definition . . . 271
A.4 Properties of $\equiv_F$ . . . 274
A.5 Sufficient Conditions for Equivalence . . . 279
A.6 Equivalence Modulo Future Updates and Reduction . . . 281
A.7 Another Formulation . . . 286
A.8 Decidability of $\equiv_F$ . . . 288
A.9 Examples . . . 289

B Confluence Proofs 291

B.1 Context . . . 291
B.2 Lemmas . . . 292
B.3 Local Confluence . . . 294
B.3.1 Local vs. Parallel Reduction . . . 295
B.3.2 Creating an Activity . . . 296
B.3.3 Localized Operations (SERVE, ENDSERVICE) . . . 297
B.3.4 Concurrent Request Sending: REQUEST/REQUEST . . 300
B.4 Case of the Calculus with *Serve*(α) . . . 301
B.5 Extension . . . 302

References 305

Index of Notations 317

Syntax of ASP Calculus 323

Operational Semantics 325

Overview of Properties 327

Overview of ASP Extensions 329

Index 337