Contents

1 **Node Replacement Graph Grammars** **1**
(J. Engelfriet, G. Rozenberg) **1**
1.1 Introduction 3
1.2 From NLC to edNCE 4
1.2.1 Node replacement and the NLC methodology 4
1.2.2 Extensions and variations: the edNCE grammar 9
1.2.3 Graph replacement grammars 14
1.2.4 Bibliographical comments 15
1.3 Node replacement grammars with Neighbourhood Controlled Embedding 16
1.3.1 Formal definition of edNCE graph grammars 16
1.3.2 Leftmost derivations and derivation trees 38
Leftmost derivations 38
Derivation trees 43
1.3.3 Subclasses 55
1.3.4 Normal forms 61
1.4 Characterizations 68
1.4.1 Regular path characterization 68
1.4.2 Logical characterization 72
1.4.3 Handle replacement 79
1.4.4 Graph expressions 81
1.5 Recognition 82
References 88

2 **Hyperedge Replacement Graph Grammars** **95**
(F. Drewes, H.-J. Kreowski, A. Habel) **95**
2.1 Introduction 97
2.2 Hyperedge replacement grammars 100
2.2.1 Hypergraphs 102
2.2.2 Hyperedge replacement 104
2.2.3 Hyperedge replacement derivations, grammars, and languages 105
2.2.4 Bibliographic notes 109
2.3 A context-freeness lemma 111
2.3.1 Context freeness 111
2.3.2 Derivation trees 114
2.3.3 Bibliographic notes 115
2.4 Structural properties 116
2.4.1 A fixed-point theorem 116

2.4.2 A pumping lemma . . . 118
2.4.3 Parikh's theorem . . . 122
2.4.4 Bibliographic notes . . . 123
2.5 Generative power . . . 124
2.5.1 Graph-generating hyperedge replacement grammars . . 124
2.5.2 String-generating hyperedge replacement grammars . . 125
2.5.3 Further results and bibliographic notes . . . 130
2.6 Decision problems . . . 132
2.6.1 Compatible properties . . . 132
2.6.2 Compatible functions . . . 135
2.6.3 Further results and bibliographic notes . . . 138
2.7 The membership problem . . . 141
2.7.1 NP-completeness . . . 141
2.7.2 Two polynomial algorithms . . . 145
2.7.3 Further results and bibliographic notes . . . 154
2.8 Conclusion . . . 155
References . . . 156

3 Algebraic Approaches to Graph Transformation - Part I: Basic Concepts and Double Pushout Approach 163
(A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, M. Löwe) **163**
3.1 Introduction . . . 165
3.2 Overview of the Algebraic Approaches . . . 168
3.2.1 Graphs, Productions and Derivations . . . 168
3.2.2 Independence and Parallelism . . . 172
Interleaving . . . 172
Explicit Parallelism . . . 174
3.2.3 Embedding of Derivations and Derived Productions . . 176
3.2.4 Amalgamation and Distribution . . . 178
Amalgamation . . . 178
Distribution . . . 179
3.2.5 Further Problems and Results . . . 180
Semantics . . . 180
Control . . . 181
Structuring . . . 182
Analysis . . . 182
More general structures . . . 182
3.3 Graph Transformation Based on the DPO Construction 182
3.4 Independence and Parallelism in the DPO approach . . . 191
3.5 Models of Computation in the DPO Approach . . . 200

3.5.1 The Concrete and Truly Concurrent Models of Computation 201
3.5.2 Requirements for capturing representation independence. 208
3.5.3 Towards an equivalence for representation independence. 212
3.5.4 The abstract models of computation for a grammar. . . 217
3.6 Embedding, Amalgamation and Distribution in the DPO approach 220
3.6.1 Embedding of Derivations and Derived Productions . . 220
3.6.2 Amalgamation and Distribution 224
Amalgamation 224
Distribution 225
3.7 Conclusion 228
3.8 Appendix A: On commutativity of coproducts 228
3.9 Appendix B: Proof of main results of Section 3.5 232
References 240

4 Algebraic Approaches to Graph Transformation - Part II: Single Pushout Approach and Comparison with Double Pushout Approach 247
(H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, A. Corradini) **247**
4.1 Introduction 249
4.2 Graph Transformation Based on the SPO Construction 250
4.2.1 Graph Grammars and Derivations in the SPO Approach 250
4.2.2 Historical Roots of the SPO Approach 258
4.3 Main Results in the SPO Approach 259
4.3.1 Parallelism 260
Interleaving 260
Explicit Parallelism 264
4.3.2 Embedding of Derivations and Derived Productions . . 268
4.3.3 Amalgamation and Distribution 273
4.4 Application Conditions in the SPO Approach 278
4.4.1 Negative Application Conditions 278
4.4.2 Independence and Parallelism of Conditional Derivations 282
Interleaving 282
Explicit Parallelism 284
4.5 Transformation of More General Structures in the SPO Approach 287
4.5.1 Attributed Graphs 288
4.5.2 Graph Structures and Generalized Graph Structures . . 292

4.5.3 High-Level Replacement Systems 294
4.6 Comparison of DPO and SPO Approach 296
4.6.1 Graphs, Productions, and Derivations 298
4.6.2 Independence and Parallelism 301
Interleaving 301
Explicit Parallelism 302
4.6.3 Embedding of Derivations and Derived Productions . . 305
4.6.4 Amalgamation and Distribution 306
4.7 Conclusion 308
References 309

5 The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic 313
(B. Courcelle) **313**
5.1 Introduction 315
5.2 Relational structures and logical languages 317
5.2.1 Structures 317
5.2.2 First-order logic 318
5.2.3 Second-order logic 319
5.2.4 Monadic second-order logic 320
5.2.5 Decidability questions 322
5.2.6 Some tools for constructing formulas 324
5.2.7 Transitive closure and path properties 327
5.2.8 Monadic second-order logic without individual variables 331
5.2.9 A worked example: the definition of square grids in MS logic 332
5.3 Representations of partial orders, graphs and hypergraphs by relational structures 334
5.3.1 Partial orders 334
5.3.2 Edge set quantifications 335
5.3.3 Hypergraphs 339
5.4 The expressive powers of monadic-second order languages . . . 340
5.4.1 Cardinality predicates 340
5.4.2 Linearly ordered structures 342
5.4.3 Finiteness 344
5.5 Monadic second-order definable transductions 346
5.5.1 Transductions of relational structures 346
5.5.2 The fundamental property of definable transductions . . 351
5.5.3 Comparisons of representations of partial orders, graphs and hypergraphs by relational structures 354

5.6 Equational sets of graphs and hypergraphs 356
5.6.1 Equational sets 356
5.6.2 Graphs with ports and VR sets of graphs 362
5.6.3 Hypergraphs with sources and HR sets of hypergraphs . 367
5.7 Inductive computations and recognizability 372
5.7.1 Inductive sets of predicates and recognizable sets 372
5.7.2 Inductivity of monadic second-order predicates 379
5.7.3 Inductively computable functions and a generalization of Parikh's theorem 383
5.7.4 Logical characterizations of recognizability 389
5.8 Forbidden configurations 390
5.8.1 Minors 390
5.8.2 The structure of sets of graphs having decidable monadic theories 396
References 397

6 2-Structures - A Framework For Decomposition And Transformation Of Graphs 401
(A. Ehrenfeucht, T. Harju, G. Rozenberg) 401
6.1 Introduction 403
6.2 2-Structures and Their Clans 404
6.2.1 Definition of a 2-structure 404
6.2.2 Clans 407
6.2.3 Basic properties of clans 410
6.3 Decompositions of 2-Structures 411
6.3.1 Prime clans 411
6.3.2 Quotients 412
6.3.3 Maximal prime clans 416
6.3.4 Special 2-structures 418
6.3.5 The clan decomposition theorem 419
6.3.6 The shape of a 2-structure 421
6.3.7 Constructions of clans and prime clans 424
6.3.8 Clans and sibas 427
6.4 Primitive 2-Structures 430
6.4.1 Hereditary properties 430
6.4.2 Uniformly non-primitive 2-structures 433
6.5 Angular 2-structures and T-structures 434
6.5.1 Angular 2-structures 434
6.5.2 T-structures and texts 438
6.6 Labeled 2-Structures 442

6.6.1 Definition of a labeled 2-structure 442
6.6.2 Substructures, clans and quotients 444
6.7 Dynamic Labeled 2-Structures 447
6.7.1 Motivation 447
6.7.2 Group labeled 2-structures 449
6.7.3 Dynamic labeled 2-structures 450
6.7.4 Clans of a dynamic Δ^δ2-structure 454
6.7.5 Horizons 455
6.8 Dynamic ℓ2-structures with Variable Domains 457
6.8.1 Disjoint union of ℓ2-structures 457
6.8.2 Comparison with grammatical substitution 458
6.8.3 Amalgamated union 459
6.9 Quotients and Plane Trees 460
6.9.1 Quotients of dynamic Δ^δ2-structures 460
6.9.2 Plane trees 461
6.10 Invariants 469
6.10.1 Introduction to invariants 469
6.10.2 Free invariants 470
6.10.3 Basic properties of free invariants 472
6.10.4 Invariants on abelian groups 473
6.10.5 Clans and invariants 476
References 476

7 Programmed Graph Replacement Systems **479**
(A. Schürr) **479**
7.1 Introduction 481
7.1.1 Programmed Graph Replacement Systems in Practice . 481
7.1.2 Programmed Graph Replacement Systems in Theory . . 482
7.1.3 Contents of the Contribution 483
7.2 Logic-Based Structure Replacement Systems 484
7.2.1 Structure Schemata and Schema Consistent Structures . 485
7.2.2 Substructures with Additional Constraints 492
7.2.3 Schema Preserving Structure Replacement 498
7.2.4 Summary 504
7.3 Programmed Structure Replacement Systems 505
7.3.1 Requirements for Rule Controlling Programs 506
7.3.2 Basic Control Flow Operators 507
7.3.3 Preliminary Definitions 510
7.3.4 A Fixpoint Semantics for Transactions 512
7.3.5 Summary 518

7.4 Context-sensitive Graph Replacement Systems – Overview . . . 519
7.4.1 Context-sensitive Graph Replacement Rules 520
7.4.2 Embedding Rules and Path Expressions 523
7.4.3 Positive and Negative Application Conditions 526
7.4.4 Normal and Derived Attributes 527
7.4.5 Summary 530
7.5 Programmed Graph Replacement Systems – Overview 530
7.5.1 Declarative Rule Regulation Mechanisms 532
7.5.2 Programming with Imperative Control Structures . . . 534
7.5.3 Programming with Control Flow Graphs 537
7.5.4 Summary 540
References 541

Index **547**