# The Geometry of Computational Obstruction

Tensor Networks, Algebraic Complexity, Mathematical Verification,
and the Continuity of Intellectual Agency

Flyxion
Independent Researcher

November 24, 2026

**Abstract**

Mathematical difficulty is not necessarily an enduring property of a proposition. The discovery of a proof transforms an unresolved search problem into an explicitly organized structure whose verification, transmission, and subsequent application may require substantially less expenditure than its original construction. This distinction provides the starting point for an investigation of computational obstruction, algebraic complexity, and the geometry of distributed dependence.

Taking the recently reported separation of VP from VNP as a principal mathematical case study, the work develops the relationship between arithmetic circuits, tensor restrictions, spectral functionals, induced pseudometrics, and nonpositive curvature. Particular attention is given to the emergence of geometrical constraints from algebraic transformation laws, the distinction between flat and contracting regimes, and the role of network decomposability in establishing computational lower bounds. The limitations of these arguments under Boolean evaluation, representational quotienting, and nonconstructive existence proofs are examined explicitly.

The mathematical analysis is extended into a general investigation of admissibility-preserving abstraction, global compatibility, resource-bounded reconstruction, and the amortization of mathematical discovery. Formal verification is distinguished from constructive accessibility, explanatory understanding, and the reproducibility of discovery processes.

Finally, the work develops an account of intellectual authorship grounded in continuity, responsibility, corrigibility, and the maintenance of publicly inspectable mathematical artifacts. Computational assistance is treated as a component of intellectual production rather than as sufficient evidence of personal agency. The independence of mathematical knowledge from its production circumstances is connected to the independence of the researcher from externally imposed urgency, compulsory interruption, and institutional control over the temporal organization of inquiry.

This version is a developmental manuscript: Chapters 1 to 49 and the appendices are developed, the reported separation remains an inherited result throughout, and the reconstructions of the inherited results are not complete.

**Keywords:** arithmetic circuit complexity; tensor restrictions; asymptotic spectrum; nonpositive curvature; formal verification; admissibility-preserving abstraction; intellectual authorship.

# Contents

**Conventions of the Formal Development** **14**

**I Mathematical Difficulty and the Completion of Knowledge** **15**

**1 The Disappearance of Mathematical Difficulty** **16**
1.1 Difficulty as a Property of an Unresolved Problem . . . . . 16
1.2 Historical Expenditure and Reusable Knowledge . . . . . 17
1.3 In What Sense Completed Mathematics Becomes P . . . . . 17
1.4 The Qualification Concerning Proof Length . . . . . 18
1.5 The Retrospective Trivialization of Discovery . . . . . 19
1.6 Formal Statements . . . . . 19

**2 Truth, Difficulty, and Explanatory Significance** **22**
2.1 Three Independent Dimensions of Achievement . . . . . 22
2.2 The Confirmation of Anticipated Results . . . . . 22
2.3 The Distinction Between Surprise and Significance . . . . . 23
2.4 General Principles Versus Isolated Resolutions . . . . . 23
2.5 Formal Statements . . . . . 24

**II The Mathematical Foundations of Computational Complexity** **26**

**3 Computational Models and Representational Commitments** **27**
3.1 Boolean Computation . . . . . 27
3.2 Arithmetic Computation . . . . . 28
3.3 Formal Polynomials and Pointwise Computation . . . . . 28
3.4 Field Dependence . . . . . 29
3.5 The Danger of Transferring Complexity Claims Between Models . . . . . 29
3.6 Formal Statements . . . . . 30

**4 Valiant's Algebraic Complexity Classes** **32**
4.1 The Definition of VP . . . . . 32

4.2 The Definition of VNP . . . 32
4.3 Local Accessibility and Global Aggregation . . . 33
4.4 Algebraic Completeness . . . 33
4.5 The Separation Problem . . . 34
4.6 Formal Statements . . . 34

**5 The Determinant, Permanent, and Hamiltonian Cycle Polynomial 36**
5.1 The Determinant . . . 36
5.2 The Permanent . . . 36
5.3 The Hamiltonian Cycle Polynomial . . . 37
5.4 Characteristic Two . . . 37
5.5 Why Superficially Similar Expressions Can Differ Computationally . . . 38
5.6 Formal Statements . . . 38

**III The Claimed Superpolynomial Separation 40**

**6 Statement, Scope, and Interpretation of the Result 41**
6.1 The Principal Lower Bound . . . 41
6.2 Asymptotic Interpretation . . . 41
6.3 What the Theorem Does Not Establish . . . 43
6.4 Exact Versus Approximate Computation . . . 43
6.5 Historical Context and Prior Lower Bounds . . . 44
6.6 Formal Statements . . . 45

**IV From Algebraic Complexity to Metric Geometry 47**

**7 Tensor Restrictions 48**
7.1 Multilinear Representations . . . 48
7.2 Tensor Products and Independent Composition . . . 48
7.3 Restrictions and Local Transformations . . . 49
7.4 Restriction as a Resource Preorder . . . 49
7.5 The Insufficiency of Individual Component Descriptions . . . 49
7.6 Formal Statements . . . 50

**8 Spectral Points and Multiplicative Invariants 51**
8.1 The Asymptotic Spectrum . . . 51
8.2 The Defining Properties of Spectral Points . . . 51
8.3 Why Multiplicativity Matters . . . 52
8.4 Nonexplicit Spectral Points . . . 52
8.5 A Plurality of Obstruction Measures . . . 52
8.6 Formal Statements . . . 53

**9 Wires, Distances, and the Emergence of Geometry** **55**
9.1 Elementary Tensor Connections . . . 55
9.2 Spectral Exponents . . . 55
9.3 Geometry as a Derived Rather Than Imposed Structure . . . 56
9.4 Network Length . . . 56
9.5 Distinguishing Spectral Length from Physical Distance . . . 56
9.6 Formal Statements . . . 57

**10 Rank, Partitions, and Hub Decompositions** **59**
10.1 Tensor Flattenings . . . 59
10.2 Rank as a Limitation on Cross-Partition Dependence . . . 59
10.3 The Two-Hub Construction . . . 59
10.4 Bottlenecks and Interface Complexity . . . 60
10.5 From Individual Cuts to Simultaneous Obstruction . . . 60
10.6 Formal Statements . . . 60

**V Curvature and the Geometry of Computational Obstruction** **62**

**11 The Crossing-Wire Identity** **63**
11.1 Crossing Dependencies . . . 63
11.2 Competing Decompositions . . . 63
11.3 The Conversion of Algebraic Identities into Geometric Inequalities . . . 64
11.4 Why Crossing Structures Matter . . . 65
11.5 Limits of the Crossing Analogy . . . 65
11.6 Formal Statements . . . 65

**12 Nonpositive Curvature** **67**
12.1 Metric Midpoints . . . 67
12.2 The Bruhat–Tits Inequality . . . 67
12.3 The Euclidean Equality Case . . . 67
12.4 CAT(0) Spaces . . . 68
12.5 Curvature as a Constraint on Computational Organization . . . 68
12.6 Formal Statements . . . 68

**13 The Flat and Contracting Alternatives** **70**
13.1 Spectral Prices . . . 70
13.2 The Contracting Hypothesis . . . 70
13.3 The Flat Hypothesis . . . 71
13.4 Exhaustiveness Without Exclusivity . . . 71
13.5 The Epistemological Significance of Unresolved Alternatives . . . 71
13.6 Formal Statements . . . 72

**VI The Flat-Case Lower Bound** **73**

**14 Complete Tensor Networks** **74**
14.1 The Complete Graph Construction . . . . . 74
14.2 The Network Polynomial . . . . . 74
14.3 The Coefficient Tensor . . . . . 75
14.4 Local Assignments and Global Compatibility . . . . . 76
14.5 Formal Statements . . . . . 76

**15 Balanced Decompositions** **78**
15.1 Balanced Partitions . . . . . 78
15.2 Circuit-Induced Tensor Decomposition . . . . . 78
15.3 The Simultaneous Spectral Inequality . . . . . 79
15.4 The Difference Between Complete Networks and Paths . . . . . 79
15.5 Why the Number of Edges Is Not Sufficient . . . . . 80
15.6 Formal Statements . . . . . 80

**16 Euclidean Configurations and the Lower-Bound Exponent** **83**
16.1 Placement of Wire Endpoints . . . . . 83
16.2 Placement of Auxiliary Hubs . . . . . 83
16.3 Geometric Savings . . . . . 84
16.4 The Resulting Lower Bound . . . . . 84
16.5 Parameter Optimization . . . . . 85
16.6 Possible Improvements . . . . . 85
16.7 Formal Statements . . . . . 85

**VII The Contracting-Case Lower Bound** **90**

**17 Contraction and Poincaré Inequalities** **91**
17.1 Unitriangular Matrix Groups . . . . . 91
17.2 Poincaré Inequalities . . . . . 91
17.3 Uniformity Across Scales . . . . . 92
17.4 Length Functions and Contraction . . . . . 92
17.5 Formal Statements . . . . . 92

**18 Duality and Rerouting Identities** **95**
18.1 From Inequalities to Identities . . . . . 95
18.2 Rerouting Laws . . . . . 95
18.3 Tree Decompositions . . . . . 96
18.4 The Algebraic Cost of Rerouting . . . . . 96
18.5 Formal Statements . . . . . 97

**19 Computations as Weighted Networks** **100**
19.1 Weighted Nondeterministic Programs . . . . . 100
19.2 Plain Instances . . . . . 100
19.3 Universal Computation . . . . . 101
19.4 Computational Histories as Tensor Networks . . . . . 101
19.5 Cutting Computations into Pieces . . . . . 101
19.6 Formal Statements . . . . . 102

**20 Accelerators and the Consequences of Small Circuits** **104**
20.1 The Universal Computation Polynomial . . . . . 104
20.2 Reduction to the Permanent . . . . . 104
20.3 Low-Degree Evaluation . . . . . 105
20.4 The Hypothetical Accelerator . . . . . 105
20.5 The Problem of Named Constants . . . . . 105
20.6 The Accumulation of Transformation Costs . . . . . 106
20.7 Formal Statements . . . . . 106

**21 Diagonalization and the Impossibility of Universal Acceleration** **108**
21.1 Self-Application . . . . . 108
21.2 Reduction to a Finite Field . . . . . 108
21.3 The Contradictory Computation . . . . . 109
21.4 The Contracting-Case Lower Bound . . . . . 109
21.5 Why the Two Cases Are Conceptually Different . . . . . 109
21.6 Formal Statements . . . . . 110

**VIII Boundaries of the Theorem** **112**

**22 Why the Argument Does Not Resolve P Versus NP** **113**
22.1 The Boolean Quotient . . . . . 113
22.2 Loss of Formal Distinctions . . . . . 113
22.3 Failure of the Balanced Decomposition Argument . . . . . 114
22.4 Failure of the Low-Degree Evaluation Argument . . . . . 114
22.5 The General Principle of Obstruction Under Quotienting . . . . . 115
22.6 Formal Statements . . . . . 115

**23 Why the Determinant Escapes the Obstruction** **118**
23.1 The Role of Completeness Reductions . . . . . 118
23.2 Structural Rather Than Superficial Differences . . . . . 118
23.3 The Location of the Proof's Failure . . . . . 119
23.4 Formal Statements . . . . . 119

**24 Border Complexity and Limiting Representations** **122**
24.1 Approximation Without Exact Realization . . . . . 122
24.2 Formal Laurent Series . . . . . 122
24.3 Singular Intermediate Quantities . . . . . 123
24.4 Persistence of Obstruction Under Limits . . . . . 123
24.5 Connections to Geometric Complexity Theory . . . . . 124
24.6 The Three Boundaries Together . . . . . 124
24.7 Formal Statements . . . . . 124

**IX From Existential Proof to Operational Knowledge** **128**

**25 The Limits of Nonconstructive Achievement** **129**
25.1 Existential Lower Bounds . . . . . 129
25.2 The Unknown Threshold . . . . . 129
25.3 Nonexplicit Obstruction Functionals . . . . . 130
25.4 Four Levels of Mathematical Accessibility . . . . . 130
25.5 The Residual Difficulty of Completed Mathematics . . . . . 131
25.6 Formal Statements . . . . . 131

**26 Mathematical Knowledge as Amortized Computation** **134**
26.1 The Cost of Discovery . . . . . 134
26.2 The Cost of Verification . . . . . 134
26.3 The Cost of Application . . . . . 135
26.4 Amortization Across Repeated Use . . . . . 135
26.5 Mathematical Civilization as Accumulated Certificates . . . . . 135
26.6 Explanation as Further Compression . . . . . 136
26.7 Formal Statements . . . . . 136

**X Formal Verification and Evidentiary Independence** **139**

**27 Proof Objects and Verification Architectures** **140**
27.1 The Distinction Between an Assertion and a Certificate . . . . . 140
27.2 Formal Proof Systems . . . . . 140
27.3 The Supplied Lean Formalization . . . . . 141
27.4 Kernel Verification Versus Semantic Verification . . . . . 141
27.5 Auditing the Circuit Model . . . . . 142
27.6 Auditing the Quantifiers . . . . . 142
27.7 Independent Replay . . . . . 143
27.8 Formal Statements . . . . . 143

**28 Verification Does Not Establish Provenance** **146**
28.1 Mathematical Validity and Production History . . . . . 146
28.2 The Non-Identifiability of Discovery Processes . . . . . 146
28.3 The Limits of Automated Verification . . . . . 147
28.4 Reproducibility of Verification Versus Reproducibility of Discovery . . . . . 147
28.5 The Independence of Mathematical Truth . . . . . 147
28.6 The Automated Manufacture of Mathematical Consensus . . . . . 148
28.7 Formal Statements . . . . . 149

**XI Authorship, Continuity, and Human Responsibility** **152**

**29 Computational Provenance Is Not Personal Authorship** **153**
29.1 The Distinction Between a Process and a Subject . . . . . 153
29.2 The Workstation Analogy . . . . . 153
29.3 Distributed Implementation . . . . . 154
29.4 Memory and Identity . . . . . 154
29.5 The Calculator Principle . . . . . 154
29.6 Attribution Without Anthropomorphism . . . . . 155
29.7 Formal Statements . . . . . 155

**30 Authorship as a Prospective Commitment** **157**
30.1 Retrospective Credit . . . . . 157
30.2 Prospective Responsibility . . . . . 157
30.3 Continuity of Accountability . . . . . 157
30.4 The Distinction Between Credit and Control . . . . . 158
30.5 Unmaintained Intellectual Artifacts . . . . . 158
30.6 Responsibility Does Not Imply Infallibility . . . . . 158
30.7 Formal Statements . . . . . 159

**31 Human Sovereignty over Computational Instruments** **161**
31.1 Delegation Without Abdication . . . . . 161
31.2 Authority Should Follow Control . . . . . 161
31.3 Human Control as an Architectural Property . . . . . 161
31.4 Intellectual Sovereignty and Inspectable Artifacts . . . . . 162
31.5 The Independence of a Maintained Exposition . . . . . 162
31.6 Formal Statements . . . . . 162

**XII Attention, Interruption, and the Temporal Organization of Intellectual Work** **164**

**32 The Economy of Imposed Urgency** **165**

32.1 The Distinction Between Importance and Urgency . . . 165
32.2 The Manufacture of Temporal Pressure . . . 165
32.3 Emotional Mobilization as an Engagement Mechanism . . . 166
32.4 Mathematics as a Publicity Event . . . 166
32.5 The Independence of Mathematical Significance from Publicity . . . 166
32.6 Formal Statements . . . 166

**33 Interruption Without Contextual Standing 168**
33.1 The Notification Problem . . . 168
33.2 The Asymmetry of Contextual Information . . . 168
33.3 The Cost of Context Switching . . . 168
33.4 The Insufficiency of Application-Local Optimization . . . 169
33.5 A Proposed Principle of Interruption Authority . . . 169
33.6 Formal Statements . . . 169

**34 Physical Pacing and Intellectual Endurance 171**
34.1 Sustainable Physical Labour . . . 171
34.2 A Simple Dynamical Model . . . 171
34.3 Local Acceleration and Global Degradation . . . 171
34.4 Intellectual Pacing . . . 172
34.5 Polemical Expenditure . . . 172
34.6 Sustainable Inquiry as a Form of Independence . . . 172
34.7 Formal Statements . . . 173

**35 Computational Dispossession and Temporal Sovereignty 174**
35.1 Material Dispossession . . . 174
35.2 Temporal Dispossession . . . 174
35.3 Attention as a Condition of Productive Capacity . . . 175
35.4 The Cost of Compulsory Reorientation . . . 175
35.5 Intellectual Independence as Control over Continuation . . . 175
35.6 Formal Statements . . . 175

**XIII Generalizing the Mathematics of Obstruction 177**

**36 Local Transitions and Global Admissibility 178**
36.1 Local Operations . . . 178
36.2 Global Compatibility . . . 178
36.3 The Complexity of Compatibility . . . 178
36.4 Tensor Representations of Admissibility . . . 179
36.5 Limits of the Representation . . . 179
36.6 Formal Statements . . . 179

**37 Gluing and the Obstruction to Independent Reconstruction** **181**
37.1 Local Sections . . . 181
37.2 Agreement on Overlaps . . . 181
37.3 Global Reconstruction . . . 181
37.4 Tensor Rank and Gluing Complexity . . . 182
37.5 Obstruction Versus Nonexistence . . . 182
37.6 Formal Statements . . . 182

**38 Admissibility-Preserving Abstraction** **184**
38.1 Quotient Maps and Lost Distinctions . . . 184
38.2 Preservation of Outputs . . . 184
38.3 Preservation of Obstruction . . . 184
38.4 The Boolean Evaluation Example . . . 185
38.5 Sufficient Abstractions for Complexity Analysis . . . 185
38.6 Formal Statements . . . 185

**39 Distributed Computation and Boundary Resources** **188**
39.1 Modular Computational Architectures . . . 188
39.2 Boundary Information . . . 188
39.3 Network Cuts and Communication Constraints . . . 188
39.4 Complete Dependency Structures . . . 189
39.5 Physical Implementation . . . 189
39.6 The Geometry of Architectural Feasibility . . . 189
39.7 Formal Statements . . . 189

**40 A Proposed General Theory of Computational Obstruction** **191**
40.1 Admissible Transformations . . . 191
40.2 Compositional Structure . . . 191
40.3 Obstruction Functionals . . . 191
40.4 Separation by Invariants . . . 192
40.5 Resource-Bounded Realization . . . 192
40.6 Geometry Induced by Admissible Transformations . . . 192
40.7 Curvature and Decomposability . . . 192
40.8 A Distinction Between Generality and Universality . . . 193
40.9 Formal Statements . . . 193

**XIV A Programme of Independent Reconstruction** **195**

**41 Establishing a Dependency Ledger** **196**
41.1 The Principal Theorem Dependencies . . . 196
41.2 Classification of Inherited Results . . . 196
41.3 Classification of Verification Status . . . 197

41.4 Recording Assumptions . . . 197
41.5 Separating Mathematical and Computational Dependencies . . . 198
41.6 Formal Statements . . . 198

**42 Elementary Worked Examples** **201**
42.1 A Two-Party Wire Tensor . . . 201
42.2 A Path Network . . . 201
42.3 A Complete Network . . . 201
42.4 Crossing Wires . . . 202
42.5 Determinant and Permanent in Characteristic Two . . . 202
42.6 Boolean Equivalence and Lost Polynomial Structure . . . 202
42.7 Verification and Discovery Costs . . . 203
42.8 Formal Statements . . . 203

**43 Computational Experiments and Reproducibility** **205**
43.1 Independent Implementations . . . 205
43.2 Exact Arithmetic . . . 205
43.3 Numerical Geometric Experiments . . . 206
43.4 Parameter Sensitivity . . . 206
43.5 Formalization Strategy . . . 206
43.6 Independent Verification Records . . . 207
43.7 Formal Statements . . . 207

**44 Maintenance, Correction, and Publication** **209**
44.1 A Maintained Mathematical Repository . . . 209
44.2 Versioned Intellectual Responsibility . . . 209
44.3 Corrections Without Concealment . . . 210
44.4 Contributions and Criticism . . . 210
44.5 The Distinction Between Maintenance and Authority . . . 210
44.6 Authorship in Practice . . . 211
44.7 Formal Statements . . . 211

**XV Conclusion: Mathematics After Discovery** **213**

**45 The Transformation of Difficulty** **214**
45.1 From Search to Verification . . . 214
45.2 Reuse and the Disappearance of the Original Search . . . 214
45.3 What the Transformation Does Not Show . . . 214
45.4 Formal Statements . . . 215

**46 Geometry as an Explanatory Language** **216**
46.1 A Common Pattern of Argument . . . . . 216
46.2 What the Pattern Explains . . . . . 216
46.3 Limits of the Language . . . . . 216
46.4 Formal Statements . . . . . 217

**47 The Independence of Mathematical Knowledge** **218**
47.1 Independence of Result and History . . . . . 218
47.2 Portability Between Environments . . . . . 218
47.3 The Conditions of Independence . . . . . 218
47.4 Formal Statements . . . . . 219

**48 The Continuity of Intellectual Agency** **220**
48.1 A Maintained Research State . . . . . 220
48.2 Instruments That Preserve Continuation . . . . . 220
48.3 The Space of Authorized Continuations . . . . . 220
48.4 Formal Statements . . . . . 221

**49 The Right Not to React** **223**
49.1 Deferral as an Operation . . . . . 223
49.2 Deferral Without Loss . . . . . 223
49.3 Batching Deferred Events . . . . . 223
49.4 A Conclusion Without Claims About Motives . . . . . 224
49.5 Formal Statements . . . . . 224

**Summary of Original Contributions** **225**

**A Mathematical Preliminaries** **227**
A.1 Conventions . . . . . 227
A.2 Matrix Rank . . . . . 227
A.3 Elementary Inequalities . . . . . 227
A.4 Formal Statements . . . . . 228

**B Arithmetic Circuit Definitions** **229**
B.1 Circuits as Lists . . . . . 229
B.2 Elementary Properties of Size . . . . . 229
B.3 Families and Classes . . . . . 229
B.4 Boolean and Approximating Variants . . . . . 230
B.5 Formal Statements . . . . . 230

**C Spectral Functionals and Tensor Restrictions** **232**
C.1 Reconstruction of the Spectral Exponents . . . . . 232
C.2 What the Reconstruction Uses . . . . . 232

C.3 Formal Statements . . . . . 232

**D Metric Geometry** **234**
D.1 Pseudometrics and Their Quotients . . . . . 234
D.2 Midpoints and the Curvature Inequality . . . . . 234
D.3 Consequences Used Later . . . . . 234
D.4 Formal Statements . . . . . 235

**E Reconstruction of the Flat-Case Proof** **236**
E.1 The Chain of Inequalities . . . . . 236
E.2 An Explicit Threshold . . . . . 237
E.3 What Remains Inherited . . . . . 237

**F Reconstruction of the Contracting-Case Proof** **239**
F.1 The Chain of Results . . . . . 239
F.2 Status of Each Step . . . . . 239
F.3 Open Obligations . . . . . 240

**G Formal Verification Audit** **241**
G.1 The Audit Register . . . . . 241
G.2 The Four Principal Propositions . . . . . 242
G.3 What a Human Reader Would Check . . . . . 243

**H Dependency Ledger** **244**
H.1 Correspondence Between Source Results and Chapters . . . . . 244
H.2 The Ledger . . . . . 245

**I Computational Experiments** **252**
I.1 Scope . . . . . 252
I.2 Experiments of Chapters 42 and 43 . . . . . 252
I.3 Output of the First Listing . . . . . 255
I.4 Crossing Identity Check . . . . . 255
I.5 Output of the Second Listing . . . . . 257

**J Correction History** **258**
J.1 Principle . . . . . 258
J.2 Corrections to Statements . . . . . 258
J.3 Typographical and Typesetting Repairs . . . . . 260

# Conventions of the Formal Development

The monograph carries a continuous mathematical spine through all of its chapters. Each chapter establishes definitions, identities, propositions, conditional results, or explicit proof obligations that later chapters inherit, so that the formulas are part of the argument and not illustrations of it. The central progression is

$$\begin{aligned}\text{Discovery} &\longrightarrow \text{Certification} \longrightarrow \text{Algebraic representation}\\ &\longrightarrow \text{Tensor restriction} \longrightarrow \text{Spectral valuation}\\ &\longrightarrow \text{Metric geometry} \longrightarrow \text{Computational obstruction}\\ &\longrightarrow \text{Independent verification} \longrightarrow \text{Maintained knowledge.}\end{aligned}$$

Four categories of mathematical statements are distinguished, and the name of the environment carries the category. A *Definition* is introduced for the monograph. A *Proposition*, *Lemma*, *Corollary*, or *Theorem* is stated together with a proof supplied here. An *Inherited Result* is a mathematical result taken from the literature or from the source manuscript, which requires citation and independent verification, and it retains that status until it has been reconstructed. A *Hypothesis*, a *Proof Obligation*, or a *Principle* is a conditional model, an unresolved obligation, or a normative proposal. The distinction matters because the monograph combines a reconstruction of an existing argument with original extensions concerning abstraction, intellectual continuity, resource expenditure, and authorship, and these should not be presented as having the same standing.

Throughout, log denotes the logarithm to base two. Inherited Results are cited by the numbering of the source manuscript [30], whose appendices are lettered. The formal spine is a proposed structure and not a completed independent proof of the separation of VP from VNP: the crossing identity, spectral duality, the curvature theorem, the machine constructions of the contracting case, the field-transfer arguments, and the border-complexity extension keep inherited status until they have been reconstructed or independently verified. Computational checks reported in the text are exact finite computations and are not proofs.

# Part I

# Mathematical Difficulty and the Completion of Knowledge

# Chapter 1

# The Disappearance of Mathematical Difficulty

## 1.1 Difficulty as a Property of an Unresolved Problem

Mathematical difficulty is usually spoken of as though it were a property of a proposition, in the way that a proposition may be true, false, or independent of a given axiom system. The history of the subject suggests a different account. A statement that resists the efforts of several generations of mathematicians may afterwards appear as a routine corollary of a general theorem, as an instance of a standard construction, or as a short calculation once an appropriate change of representation has been made. Fermat's assertion about the equation $x^n + y^n = z^n$ stood unresolved for more than three centuries and is now understood as a consequence of the modularity of elliptic curves over the rationals, a statement whose proof draws on machinery that the original problem gave no reason to anticipate. The question whether primality can be decided in deterministic polynomial time, open for decades in spite of a sustained programme of work on randomized and conditional algorithms, was settled by an argument short enough to be taught in a single lecture [2]. In neither case did the proposition change. What changed was the representational environment in which it could be approached.

It is therefore useful to treat difficulty as a relation between a proposition and a state of mathematical knowledge, rather than as an intrinsic attribute of the proposition alone. Relative to an impoverished stock of concepts, techniques, and notations, a statement may offer no accessible path toward a proof. Relative to an enriched stock, the same statement may be reached in a few steps. The distinction that matters for the present essay is between two different expenditures. The first is the expenditure required to discover a mathematical structure, which includes the selection of definitions, the identification of the correct generalization, and the elimination of approaches that fail. The second is the expenditure required to navigate that structure once it has been made explicit, which consists in following definitions, applying lemmas, and carrying out verifications whose form is fixed in advance. The former is typically enormous and irregular. The latter is typically modest

and regular. A proof is the document in which the second kind of expenditure is substituted for the first.

## 1.2 Historical Expenditure and Reusable Knowledge

The distinction can be sharpened by separating the historical expenditure required to obtain a result from the expenditure required to reproduce, understand, or apply it. The historical expenditure includes the failed conjectures, the abandoned lines of attack, the computational searches that produced nothing, and the circumstances of institutional support, collaboration, and chance that shaped the order in which ideas were tried. A published proof preserves almost none of this. It preserves the successful organization of reasoning, stripped of the unsuccessful investigations that preceded it, and presents that organization in an order determined by logical dependence rather than by the order of discovery. This loss is not a defect. It is precisely what makes the result reusable, since a reader who is required to retrace the entire history of a discovery has gained nothing over the original discoverer.

Mathematical progress may accordingly be understood as the conversion of historical discovery costs into reusable structures. A theorem, together with its proof and the definitions on which it relies, is a durable object that can be copied, transmitted, checked, taught, and applied at a marginal cost that is independent of the cost of its creation. The amortization of that creation cost over later uses will be given an explicit form in a later chapter. For the present, the observation is only that the economics of mathematical knowledge differ from the economics of its production. A result that required decades of effort may be verified in an afternoon and invoked in a single line, and this asymmetry is what allows mathematical knowledge to accumulate across generations rather than being rediscovered by each.

## 1.3 In What Sense Completed Mathematics Becomes P

The informal claim that completed mathematics becomes easy has a precise counterpart in the vocabulary of computational complexity, though it requires care to state correctly. Let $T$ denote a proposition, encoded as a finite string in some fixed formal language, and let $\pi$ denote a candidate certificate of its truth, that is, a finite string intended to be a proof of $T$ in some fixed proof system. Associated with the proof system is a verification relation

$$V(T, \pi) \in \{0, 1\},$$

which takes the value 1 exactly when $\pi$ is a correct proof of $T$ in the system. For a proof system in which derivations are written out explicitly in a fixed calculus, the verification relation is computable in time bounded by a polynomial in the combined length of its arguments, so that

$$\mathrm{Time}\big(V(T, \pi)\big) \;\leq\; p\big(|T| + |\pi|\big)$$

for some polynomial $p$. The discovery problem is a different computational task, namely

$$\text{find } \pi \text{ such that } V(T, \pi) = 1.$$

Once a certificate $\pi$ is in hand, the original search is no longer required, and the remaining task is the evaluation of $V$ on a specific pair of strings. The difference between the two tasks is the same difference that defines the class NP, in which membership is witnessed by a certificate that can be checked efficiently even when no efficient method for producing it is known [4].

**Principle 1.1** (Certificate-Conditioned Tractability)**.** Fix a proof system whose verification relation $V$ is decidable in time polynomial in $|T| + |\pi|$. For every proposition $T$ for which a certificate $\pi$ is available, the task of confirming that $T$ holds is reduced from the search problem of finding $\pi$ to the evaluation of $V(T, \pi)$, and the cost of this evaluation is polynomial in the lengths of $T$ and $\pi$.

The principle concerns the computational task faced by a party who possesses a certificate, and it is conditional on that possession. It is a statement about the cost of verification, not about the cost of discovery, and it does not assert that the set of proven propositions coincides with the complexity class P. Such an identification would be false as stated, because the proven propositions are not a language of decision problems of the kind P classifies, and because the complexity of any single proposition with a fixed certificate is a constant. What the principle captures is the transformation of the problem faced by a reader: before a certificate exists the task is a search, and afterward it is a check. In this restricted and conditional sense completed mathematics becomes tractable.

## 1.4 The Qualification Concerning Proof Length

The tractability asserted in the previous section is polynomial in the combined length $|T| + |\pi|$, and this is not the same as being polynomial in the length of the proposition $|T|$ alone. A proof may itself be exponentially larger than the statement it establishes, in which case verification, although efficient in the length of the certificate, is not efficient in the length of the statement. Whether proofs of polynomial length exist for all members of a natural class of statements is a substantive question and not a matter of definition. Cook and Reckhow showed that the existence of a propositional proof system in which every tautology has a proof of length polynomial in the tautology is equivalent to the assertion that NP equals coNP [12]. Strong lower bounds are known for restricted systems. In particular, Haken proved that resolution refutations of the pigeonhole principle require exponential size [18], so that for that system verification is polynomial in the size of the proof while the proofs themselves cannot be short.

This qualification is essential to the integrity of the argument. If completed mathematics were asserted to be tractable in the length of the proposition, the epistemological claim would imply the existence of short proofs for large classes of statements, and would thereby settle

precisely the complexity-theoretic questions whose investigation is the central subject of this essay. The complexity of checking a proof also depends on the proof system and on the representation in which the certificate is written, since the same argument may be presented as a natural-language text, as a derivation in a sequent calculus, or as a term in a dependent type theory, with different costs of checking in each. The principle of certificate-conditioned tractability is therefore to be read as relative to a fixed proof system and a given certificate, and every later use of it will carry that relativization.

## 1.5 The Retrospective Trivialization of Discovery

Mathematicians are familiar with the phenomenon whereby an elegant proof makes a result that was previously forbidding appear inevitable. After the argument has been seen, it is difficult to recover the state of mind in which the statement was opaque, and the reader is inclined to regard the earlier difficulty as an accident of insufficient cleverness. This impression is partly an illusion produced by hindsight, but it also reflects something real about the function of proof. The best mathematical explanations do not merely establish propositions. They reorganize the representational environment so that the propositions become natural consequences of accessible principles, and in doing so they alter the effort that any later reader must expend.

This suggests that mathematical understanding is partly the systematic elimination of unnecessary search. A definition that isolates the correct notion, a lemma that factors a recurring step, and a change of variables that renders a computation transparent each remove a region of a search space that earlier workers had to traverse. What remains after these reductions is a structure in which the passage from hypotheses to conclusion is guided, so that a reader who follows it does not need to guess. The disappearance of difficulty that accompanies a completed proof is thus the visible trace of a transfer of labor from the reader to the structure itself.

## 1.6 Formal Statements

**Definition 1.2** (Verification Relation)**.** Let $\Sigma$ be a finite alphabet, $\mathcal{S} \subseteq \Sigma^*$ a space of encoded statements, and $\mathcal{P} \subseteq \Sigma^*$ a space of encoded proofs. A verification relation is a function $V : \mathcal{S} \times \mathcal{P} \to \{0,1\}$. The set of valid certificates for a statement $T$ is $\mathcal{C}(T) = \{\pi \in \mathcal{P} : V(T,\pi) = 1\}$.

**Definition 1.3** (Discovery Problem)**.** Given $T \in \mathcal{S}$, Discover$(T)$ is the task of producing an element of $\mathcal{C}(T)$. It is a search problem over the space of admissible certificates.

**Definition 1.4** (Minimal Certificate Length)**.** $\ell_V(T) = \inf_{\pi \in \mathcal{C}(T)} |\pi|$, with $\ell_V(T) = \infty$ when $\mathcal{C}(T) = \varnothing$.

**Proposition 1.5** (Certificate-Conditioned Tractability)**.** *Suppose $V$ is computable in time polynomial in $|T| + |\pi|$. Then the language $L_V = \{(T,\pi) : V(T,\pi) = 1\}$ belongs to* P.

*The projected language* $L_{\exists V} = \{T : \exists \pi,\ V(T, \pi) = 1\}$ *need not belong to* P*. If there is a polynomial* $q$ *with* $\ell_V(T) \leq q(|T|)$ *for every* $T \in L_{\exists V}$*, then* $L_{\exists V} \in$ NP*.*

*Proof.* The first statement is the assumed complexity of $V$. For the second, take $V$ to check derivations in a standard proof calculus for first-order logic; this is polynomial-time computable, while $L_{\exists V}$ is the set of valid sentences, which is undecidable by Church's theorem and therefore not in P. For the third, the polynomially bounded witnesses and the polynomial-time verifier $V$ give the definition of NP stated in Chapter 3. □

This is the precise formal interpretation of Principle 1.1.

**Proposition 1.6** (Elimination of Repeated Search)**.** *For a supplied valid certificate* $\pi$ *of* $T$*, subsequent verification costs* $C_{\text{reuse}}(T, \pi) = C_V(T, \pi)$*, whereas an initial search-and-verification procedure costs* $C_{\text{initial}} = C_D(T) + C_V(T, \pi)$*.*

The proposition concerns a specified accounting model, in which a party who holds a certificate pays only for verification. It does not assert that verification is always inexpensive in absolute terms.

**Definition 1.7** (Knowledge Environment)**.** A knowledge environment $K$ is a specification of the certificates available to a given agent, together with an accessibility predicate $A_K(T, \pi) \in \{0, 1\}$ recording whether the agent can retrieve $\pi$ and run the checker for $V$ on it. Accessibility is a resource condition: a certificate may exist without being accessible because it is unpublished, expensive to retrieve, dependent on unavailable software, or encoded in an inaccessible representation.

**Definition 1.8** (Effective Reconstruction Cost)**.** $C_K(T) = \inf\{C_{\text{access}}(\pi) + C_V(T, \pi) : V(T, \pi) = 1,\ A_K(T, \pi) = 1\}$, with the infimum over the empty set equal to $\infty$.

**Proposition 1.9** (Existence Without Availability)**.** $\mathcal{C}(T) \neq \varnothing$ *does not imply* $C_K(T) < \infty$*, and* $C_K(T) \geq \inf_{\pi \in \mathcal{C}(T)} C_V(T, \pi)$*.*

*Proof.* If every valid certificate is inaccessible in $K$, the infimum defining $C_K(T)$ ranges over the empty set. When it is nonempty, it is a restriction of the infimum of $C_V$ over $\mathcal{C}(T)$ to fewer certificates with the nonnegative term $C_{\text{access}}$ added, hence at least as large. □

**Proposition 1.10** (Theorems as a Language)**.** *Once an encoding is fixed, the set* $\text{Thm} = L_{\exists V}$ *of provable statements is a language. It may be undecidable and in particular outside* P*.*

Two qualifications apply to Proposition 1.5. First, the polynomial bound on $C_V$ concerns an explicit certificate in a chosen calculus. Proof assistants may employ dependent type checking, definitional equality, normalization, and computational reflection, so that a short proof script can initiate a computation whose cost is not polynomial in the length of the script, and the script and the explicit derivation it generates must be distinguished. Second, the statement that a certificate is reused at a cost independent of its creation cost is an accounting convention in the model above and not a theorem, since discovery and verification

costs need not be statistically or computationally independent. A certificate establishes derivability relative to a proof system and does not by itself establish the soundness of that system or truth in an intended interpretation.

# Chapter 2

# Truth, Difficulty, and Explanatory Significance

## 2.1 Three Independent Dimensions of Achievement

The achievement represented by a mathematical result is not a single quantity. At least three dimensions should be distinguished. The first is the truth of the result, which is settled once a correct proof is available. The second is the difficulty of discovering that proof, which measures the expenditure that was required of those who found it. The third is the explanatory power of the resulting structure, which measures how much of the surrounding mathematical landscape the proof illuminates and how readily its methods extend to other questions. These dimensions vary independently. A technically formidable proof of an expected proposition may carry less conceptual novelty than a comparatively elementary proof that introduces a new invariant, and an expected result may have very great importance if its proof establishes a general method that had previously been unavailable.

The point has consequences for the evaluation of any announced result. Difficulty, whether measured by the number of years a problem was open or by the number of capable researchers who failed to resolve it, is evidence about the first kind of expenditure and tells little about the third. A proposition may be hard to prove because it lies at the intersection of several poorly understood areas, in which case its resolution may yield a great deal of understanding, or because it demands a long and intricate verification of many cases, in which case its resolution may yield very little. The assessment of significance therefore requires inspection of the structure of the proof, and cannot be read from the history of failed attempts.

## 2.2 The Confirmation of Anticipated Results

Two separations dominate the landscape of computational complexity. The first is the expectation that $\mathrm{P} \neq \mathrm{NP}$, which asserts that the efficient verification of certificates does not imply their efficient discovery. The second is the analogous expectation in the algebraic

setting introduced by Valiant [34], namely

$$\mathrm{VP} \neq \mathrm{VNP},$$

which asserts that polynomials expressible as exponentially large sums of efficiently computable terms need not themselves be efficiently computable by arithmetic circuits. Both expectations are held with considerable confidence by the community of researchers who work on them, and both rest on a mixture of heuristic arguments, failed attempts at efficient algorithms, and the structure of known reductions.

The intuitive plausibility of a separation is not equivalent to a proof of it. Nevertheless, proving a separation that was anticipated differs epistemologically from discovering that the prevailing conceptual picture was mistaken. In the former case the result confirms a belief and converts it into knowledge, and its significance lies in the methods by which this was accomplished and in the consequences those methods have for neighbouring questions. In the latter case the result overturns a belief, and its significance lies in the revision it forces. A discussion of a claimed separation that conflates these two possibilities risks attributing to the result a degree of surprise that it does not have, or withholding from it an importance that it does.

## 2.3 The Distinction Between Surprise and Significance

A mathematical result need not be surprising to be significant, and a surprising result need not be mathematically profound. Surprise is a relation between a result and the prior expectations of an audience, and it can be produced by circumstances unrelated to the depth of the mathematics, including the form in which the result is announced, the reputation of those who announce it, and the speed with which the announcement circulates. Significance is a relation between a result and the body of mathematics to which it belongs, and it is determined by what the result explains, what it makes possible, and what it connects.

It follows that the intensity of an audience's immediate emotional response is not a reliable measure of intellectual significance, and the identification of the two should be rejected. A sober assessment of a result asks what the proof establishes, by what means, under what assumptions, and with what consequences, and it is indifferent to whether the answer is greeted with astonishment or with a shrug. The appropriate attitude toward an announcement is the attitude one takes toward any mathematical claim: that its content is to be examined, and that its reception is a separate fact about the community that received it.

## 2.4 General Principles Versus Isolated Resolutions

The evaluative question that will guide the remainder of this essay may now be stated. Given a proof of a named conjecture, does it merely resolve that conjecture, or does it expose a reusable structure that explains a wider class of phenomena? A proof of the first kind adds

one proposition to the stock of known truths. A proof of the second kind adds a method, a concept, or a viewpoint, from which many propositions, including some not yet formulated, may be derived. The difference is analogous to the difference between a table entry and the formula that generates the table.

In the case of the reported separation of VP from VNP, the principal candidate for a reusable structure is the relationship between tensor restrictions, metric geometry, network decomposition, and computational obstruction. The later parts of this essay reconstruct that relationship in detail, with the aim of separating what the argument establishes about arithmetic circuits from what it suggests about the organization of dependencies more generally. The distinction drawn here, between a resolved conjecture and an exposed structure, will serve as the criterion by which the mathematical material is assessed, and it will also govern the choice of which parts of the argument deserve extended treatment.

## 2.5 Formal Statements

**Definition 2.1** (Mathematical Achievement Profile)**.** A mathematical result is represented by the tuple $\mathfrak{M}(T,\pi) = \big(V(T,\pi), C_D, C_V, C_A, \ell_V, G\big)$, where $G$ denotes a specified measure of generality or explanatory reach. No canonical scalar measure of $\mathfrak{M}$ is assumed.

**Definition 2.2** (Relative Derivational Complexity)**.** For a background theory $B$ and a fixed proof calculus, $K_B(T) = \min\{|\pi| : B \vdash_\pi T\}$, with the minimum over the empty set equal to $\infty$.

**Proposition 2.3** (Monotonicity Under Enrichment)**.** *If $B \subseteq B'$ in a common proof calculus, then $K_{B'}(T) \le K_B(T)$.*

*Proof.* Every derivation available from $B$ remains a derivation from $B'$. □

**Corollary 2.4** (Retrospective Simplification)**.** *If a previously established theorem $S$ is added as an available premise, then $K_{B\cup\{S\}}(T) \le K_B(T)$.*

This formalizes how accumulated mathematical knowledge can reduce subsequent derivational complexity, while excluding the original cost of establishing the additional premise.

**Definition 2.5** (Truncated Derivational Complexity)**.** For a budget $M < \infty$ let $K_B^{(M)}(T) = \min\{K_B(T), M\}$. Each quantity in the achievement profile is understood relative to a specified algorithm, computational model, and background theory; discovery expenditure in particular may denote the historical expenditure, the least expenditure among known procedures, or a hypothetical minimum, and these are different quantities.

**Hypothesis 2.6** (Explanatory Compression)**.** *For a finite family $\mathcal{F}$ of statements and a premise $S$, define*

$$\Delta_B^{(M)}(\mathcal{F}; S) = \sum_{T\in\mathcal{F}} \big(K_B^{(M)}(T) - K_{B\cup\{S\}}^{(M)}(T)\big),$$

*which is a finite sum of finite terms. The net explanatory value of $S$ is $\Delta_B^{(M)}(\mathcal{F}; S) - C_{\text{intro}}(S)$, where $C_{\text{intro}}(S)$ is the cost of introducing, establishing, storing, and maintaining $S$. Without the subtracted term the optimal strategy would be to adopt every proposition of interest as an available premise.*

**Proposition 2.7** (Amortized Cost of Shared Knowledge)**.** *Let a theorem have discovery cost $D$, initial verification cost $V$, infrastructure cost $I$, maintenance cost $M(N)$ over $N$ uses, and application costs $A_1, \ldots, A_N$. The mean cost per use is*

$$\overline{C}(N) = \frac{D + V + I}{N} + \frac{M(N)}{N} + \frac{1}{N} \sum_{i=1}^{N} A_i.$$

*If $M(N) = o(N)$ and $\frac{1}{N} \sum_{i \le N} A_i \to \bar{A} < \infty$, then $\overline{C}(N) \to \bar{A}$. If $M(N) \ge \mu N$ for some $\mu > 0$, then $\overline{C}(N) \ge \mu$ for all $N$ and the fixed costs never become negligible relative to maintenance.*

*Proof.* The identity is the definition of the mean. The first term tends to zero as $N \to \infty$ for fixed $D + V + I$, the second tends to zero when $M(N) = o(N)$, and the third converges by assumption. For linear maintenance the second term is at least $\mu$ and the others are nonnegative. □

The proposition shows why durability is itself an achievement: knowledge becomes inexpensive per use only where the structures that prevent repeated discovery remain accessible, and proofs do not remain equally accessible under arbitrary changes in software dependencies, notation, background theories, or institutions.

# Part II

# The Mathematical Foundations of Computational Complexity

# Chapter 3

# Computational Models and Representational Commitments

## 3.1 Boolean Computation

A decision problem is a set $L \subseteq \{0,1\}^*$ of finite binary strings, and to solve it is to decide, for an arbitrary string $x$, whether $x \in L$. The two standard formalizations of the process of deciding are the Turing machine and the Boolean circuit. A Turing machine is a finite control equipped with a tape, and its cost on an input is the number of steps before it halts. A Boolean circuit is a directed acyclic graph whose internal nodes are AND, OR, and NOT gates and whose leaves are input bits, and its cost is the number of gates. A single circuit handles inputs of one length, so that a language is decided by a family of circuits $(C_n)$, one for each length $n$. The class P consists of the languages decided by a deterministic Turing machine in time bounded by a polynomial in the length of the input. The class NP consists of the languages $L$ for which there exist a polynomial $q$ and a relation $R$ decidable in polynomial time such that $x \in L$ if and only if some string $w$ of length at most $q(|x|)$ satisfies $R(x, w) = 1$.

Two commitments in these definitions deserve explicit attention because they are easily forgotten. The first concerns encoding. The statement that an algorithm runs in polynomial time presupposes a convention for the length of the input, and a change of convention, such as writing an integer in unary rather than in binary, can alter a problem from intractable to tractable without any change in the underlying mathematical object. The second concerns uniformity. A family of Boolean circuits $(C_n)$ may be nonuniform, in the sense that nothing requires the circuit for length $n + 1$ to be related in any computable way to the circuit for length $n$, and the class of languages decided by nonuniform families of polynomial size is conventionally written P/poly. This class contains P and also contains undecidable languages, so that statements about it are statements about the existence of small circuits rather than about the existence of algorithms for finding them. The definition of NP makes the distinction between efficient discovery and efficient verification central: membership is witnessed by a certificate that can be checked quickly, and nothing is asserted about how

quickly the certificate can be found. The discussion of certificates in Chapter 1 is an instance of exactly this structure, with a proof in the role of the witness.

## 3.2 Arithmetic Computation

The arithmetic circuit replaces the Boolean connectives by the operations of a ring. Following the convention of the source manuscript [30], an arithmetic circuit over a commutative ring $R$ is a finite list of polynomials over $R$, written

$$g_1, \ldots, g_L,$$

in which each entry $g_i$ is a variable, a constant from $R$, or the sum or the product of two earlier entries, which may be the same entry. The entries are the gates of the circuit, the circuit is said to compute each polynomial in the list, and its size is the length $L$ of the list, so that variables and constants are counted as gates. Counting only the additions and multiplications changes the size by a factor of at most three, which is immaterial for the asymptotic statements considered here. No division is available. Subtraction is obtained from addition and multiplication by the constant $-1$, so that the model computes exactly the polynomials that can be built from the variables and the constants of $R$ by additions and multiplications.

The model differs from a Boolean circuit in a feature that will matter throughout. An arithmetic circuit is required to compute a polynomial, which is a formal object, and not merely a function on a restricted set of inputs. Correctness as a formal polynomial entails correctness at every element of every extension of the field, since an identity between polynomials is an identity in every extension, but this is a consequence of the formal identity and not a separate evaluation requirement imposed on each circuit. The circuit is also permitted to reuse a computed value any number of times, and this distinguishes circuits from formulas. In a formula each intermediate result is used at most once, so that the underlying graph is a tree. The polynomial $x^{2^k}$ is computed by $k+1$ gates when each square is formed once and reused, whereas a formula for it requires on the order of $2^k$ occurrences of the variable. The possibility of reuse is what allows circuits to represent polynomials of exponential degree and, as Chapter 5 shows, polynomials with factorially many monomials, by compact expressions.

## 3.3 Formal Polynomials and Pointwise Computation

To compute a formal polynomial is a stronger requirement than to reproduce its values on a restricted set of inputs, and the gap is visible already in one variable. The polynomial

$$x^2 - x$$

vanishes at both Boolean inputs, 0 and 1, and nevertheless is not the zero polynomial, since it has degree two. More generally, two polynomials may agree on every point of $\{0,1\}^n$ while possessing quite different algebraic structures. Over any field, a function on $\{0,1\}^n$ is represented by a unique multilinear polynomial, namely one in which no variable occurs with an exponent greater than one, and two polynomials agree on the cube exactly when their difference lies in the ideal generated by the polynomials $x_i^2 - x_i$. The ideal therefore measures what is lost when polynomials are replaced by their Boolean evaluations, and a large part of Chapter 22 will turn on the identification it induces.

The same phenomenon occurs over finite fields without any restriction to Boolean inputs. Over the field with $q$ elements, the polynomial $x^q - x$ vanishes at every point and is still not the zero polynomial. The formal model of the arithmetic circuit retains these distinctions, and so a lower bound for the formal computation of a polynomial does not by itself constrain the computation of any other polynomial that happens to have the same values on a given set of points.

## 3.4 Field Dependence

An arithmetic circuit is defined over a field, and the field determines which constants are available and which identities hold. In characteristic zero, the integers embed in the field without collapse, and constants such as $\frac{1}{2}$ are available. In positive characteristic $p$ the integer $p$ is zero, so that $p$ copies of the same quantity sum to nothing, and the elements $\pm\frac{1}{2}$ do not exist when $p = 2$. A circuit over a field $k$ is also a circuit over every extension of $k$, which is the reason that a circuit over the reals must handle complex inputs as well. Conversely, a lower bound that holds over a field holds over any subfield, since a circuit over the subfield is a circuit over the field.

The formal algebraic model retains distinctions that disappear under evaluation on a finite field or on a subset of the field. Chapters 5 and 23 will show how field characteristic bears on the relationship between the permanent and the determinant, and the manuscript's own analysis, summarized in Chapter 6, observes that in its argument the field supplies the scalars of the tensors and the linear maps that restrict them, and that the only property of the field used in the reduction to the permanent is that $2 \neq 0$ [30]. Which of two technical hypotheses on which the argument branches holds is itself reported to depend on the field. These dependencies will be tracked explicitly in the ledger of Chapter 41, since a statement that is uniform in the field and a statement with a field-dependent threshold are different mathematical claims.

## 3.5 The Danger of Transferring Complexity Claims Between Models

The preceding sections suggest a general principle that will be used repeatedly. A lower bound is always a statement about a cost relative to a model, and it constrains computations

in that model and in any model whose computations can be converted into it at controlled expense. It says nothing about a model that permits operations the first does not.

**Principle 3.1** (Model-Relative Obstruction)**.** Let $M$ and $M'$ be models of computation with cost functions $c$ and $c'$, and suppose that every computation of a quantity $x$ in $M'$ at cost $s$ can be converted into a computation of $x$ in $M$ at cost at most $\theta(s)$. If every computation of $x$ in $M$ costs more than $\theta(s)$, then every computation of $x$ in $M'$ costs more than $s$. No conclusion about $M'$ follows if the conversion is unavailable or if $M'$ identifies objects that $M$ distinguishes.

The principle can be illustrated by the relationship between Boolean and arithmetic circuits. An arithmetic circuit over the field with two elements converts into a Boolean circuit of comparable size, since addition is exclusive disjunction and multiplication is conjunction, and the resulting circuit computes the same function on Boolean inputs. A lower bound for Boolean circuits therefore implies a lower bound for arithmetic circuits over that field, which is the direction in which implications between the two settings are known, subject to hypotheses on the field [11]. The converse direction requires care. A Boolean circuit can be arithmetized with constant-factor overhead in the number of gates, using $\mathrm{AND}(x, y) = xy$, $\mathrm{NOT}(x) = 1 - x$ and $\mathrm{OR}(x, y) = x + y - xy$, and the result reproduces the Boolean behavior on $\{0, 1\}^n$ over every field. What fails is the preservation of a prescribed formal polynomial: the arithmetized circuit computes some polynomial that agrees with the target on the Boolean cube, that is, one that differs from it by an element of the kernel of the evaluation map, and it need not compute the target polynomial itself. The obstruction is a failure of efficient lifting through a quotient. For this reason a superpolynomial lower bound for arithmetic circuits does not transfer to Boolean circuits, and the manuscript states plainly that its result does not settle the Boolean problem P versus NP [30]. The principle will explain, in Chapter 22, precisely where the argument ceases to apply.

## 3.6 Formal Statements

**Definition 3.2** (Circuit Complexity)**.** Over a field $\mathbb{F}$ let $\mathcal{A} = \mathbb{F}[x_1, \ldots, x_n]$. For $f \in \mathcal{A}$ let $L_{\mathbb{F}}(f) = \min\{|C| : C \text{ computes } f\}$, the minimum size of an arithmetic circuit $C = (g_1, \ldots, g_L)$ that computes $f$ in the sense of Section 3.2.

**Proposition 3.3** (Elementary Composition Bounds)**.** $L_{\mathbb{F}}(f + g) \le L_{\mathbb{F}}(f) + L_{\mathbb{F}}(g) + 1$ *and* $L_{\mathbb{F}}(fg) \le L_{\mathbb{F}}(f) + L_{\mathbb{F}}(g) + 1$.

*Proof.* Concatenate the two circuits and append one gate that adds or multiplies the two outputs. □

**Definition 3.4** (Boolean Evaluation Map)**.** $E : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}^{\{0,1\}^n}$, $E(f)(a) = f(a)$.

**Proposition 3.5** (Loss of Formal Information)**.** $\ker E = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$, *and consequently* $\mathbb{F}[x_1, \ldots, x_n]/\ker E \cong \mathbb{F}^{\{0,1\}^n}$.

*Proof.* Each generator vanishes on $\{0,1\}^n$, so the ideal $I$ is contained in the kernel. Conversely, let $f \in \ker E$. Replacing $x_i^2$ by $x_i$ repeatedly produces a multilinear polynomial $\tilde{f}$ with $f - \tilde{f} \in I$, and $E(\tilde{f}) = E(f) = 0$. A multilinear polynomial vanishing on $\{0,1\}^n$ is zero, by induction on $n$: write $\tilde{f} = g + x_n h$ with $g, h$ multilinear in $n-1$ variables. Setting $x_n = 0$ shows that $g$ vanishes on $\{0,1\}^{n-1}$, so $g = 0$; setting $x_n = 1$ then shows that $h$ vanishes there, so $h = 0$. Hence $\tilde{f} = 0$ and $f \in I$. The map $E$ is surjective, since the indicator of a point $a \in \{0,1\}^n$ is the image of $\prod_i (x_i \text{ or } 1 - x_i)$ according to $a_i$, and the isomorphism follows. □

This quotient becomes central to Chapter 22.

**Proposition 3.6** (Arithmetization and the Failure of Lifting)**.** *Over any field, a Boolean circuit with $s$ gates over $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$ is converted, with at most a constant factor overhead, into an arithmetic circuit whose evaluation map image equals the Boolean function computed. If the Boolean circuit computes $E(f)$ for a polynomial $f$, the arithmetic circuit computes some $g$ with $g - f \in \ker E$, and $g = f$ need not hold.*

*Proof.* Replace each gate by the polynomial $xy$, $x + y - xy$, or $1 - x$ respectively, and each input by a variable. These agree with the gates on $\{0,1\}$, so $E(g) = E(f)$, and Proposition 3.5 gives $g - f \in \ker E$. For $f = x_1$ and $g = x_1^2$ the evaluations agree and the polynomials differ. □

**Definition 3.7** (Quotient Cost)**.** Let $q : X \to Y$ be surjective and $C_X : X \to [0, \infty]$ a cost of realizing elements of $X$. Define $C_q(y) = \inf_{x:\, q(x)=y} C_X(x)$ and, when finite, the lost distinction $\mathcal{L}_q(x) = C_X(x) - C_q(q(x))$.

**Proposition 3.8** (Abstraction Cannot Increase Cost)**.** *$C_q(q(x)) \le C_X(x)$ and $\mathcal{L}_q(x) \ge 0$ whenever finite. If there is a lift $\ell : Y \to X$ with $q \circ \ell = \mathrm{id}_Y$ and $C_X(\ell(y)) \le \theta(C_q(y))$ for a function $\theta$, then every $x$ in the image of $\ell$ satisfies $C_X(x) \le \theta(C_q(q(x)))$.*

*Proof.* The first claim holds because $x$ is one of the representatives over which the infimum is taken. For $x = \ell(y)$ we have $q(x) = y$ and $C_X(x) \le \theta(C_q(y))$ by hypothesis. □

For $X = \mathbb{F}[x_1, \ldots, x_n]$, $Y = \mathbb{F}^{\{0,1\}^n}$, $q = E$ and $C_X = L_{\mathbb{F}}$, one obtains $L_{\mathrm{eval}}(f) \le L_{\mathbb{F}}(f)$, which explains why a lower bound for the formal complexity of $f$ does not give the same lower bound for computing its Boolean evaluations: a cheaper representative may lie in the same class. The critical condition for transferring a lower bound is not preservation of the output but the existence of a cost-controlled lift.

# Chapter 4

# Valiant's Algebraic Complexity Classes

## 4.1 The Definition of VP

Valiant introduced algebraic analogues of P and NP in which the objects of study are families of polynomials [34]. A family $(f_n)_{n\geq 1}$ of polynomials over a field belongs to VP if the number of variables of $f_n$, the degree of $f_n$, and the size of the smallest arithmetic circuit that computes $f_n$ are all bounded by polynomials in $n$. The emphasis on families is essential, since every individual polynomial has a circuit, and a statement about efficiency has content only as a statement about the growth of a sequence. The bound on the degree is equally essential. Without it the family $x^{2^n}$, which has circuits of size $n+1$, would be counted as efficiently computable although its degree is exponential in $n$, and the class would cease to capture the intuition of polynomial-size computation. The degree condition ensures that the polynomial itself, and not only its circuit, is of moderate size.

## 4.2 The Definition of VNP

The class VNP is defined from VP by summation over the Boolean cube. A family $(f_n)$ belongs to VNP if there is a family $(g_n)$ in VP and a polynomially bounded function $r$ such that

$$f_n(x) = \sum_{e \in \{0,1\}^{r(n)}} g_n(x, e),$$

where the variables $e$ range over the Boolean cube and are substituted into $g_n$ as numbers. Taking $r(n) = 0$ shows that VP $\subseteq$ VNP, and the conjecture that the inclusion is proper is Valiant's conjecture. The definition mirrors the existential quantifier in the definition of NP: a certificate $e$ is chosen from a set of exponential size, and a property of the pair $(x, e)$ is evaluated efficiently, with the difference that in the algebraic setting the certificates are summed rather than searched.

The structure to be noted is the distinction between the cost of an individual summand and the cost of the aggregate. Each term $g_n(x, e)$ is available at polynomial cost, by the hypothesis that $(g_n)$ is in VP. The sum contains $2^{r(n)}$ such terms, and unless some cancellation or regularity among them can be exploited, computing the sum by enumeration is exponential. Whether such regularity always exists is exactly the question whether VNP collapses into VP.

## 4.3 Local Accessibility and Global Aggregation

The class VNP is defined by algebraic summation and not by existential quantification, and the analogy with NP is correspondingly loose: in positive characteristic contributions may cancel, so that a vanishing value does not indicate the absence of compatible assignments. The class is better read as efficiently specified but potentially expensive global aggregation than as the discovery of a witness. The definition of VNP is an instance of a more general structural problem, namely that locally accessible contributions need not admit efficient global aggregation. Each term of the sum is cheap, the individual contributions are indexed by objects that are easy to describe, and the difficulty lies entirely in combining exponentially many of them. The manuscript's own polynomials have this form. Both of the polynomials that it uses in its two cases are described as clause sums, that is, sums over all bit vectors of products of local weights, so that each factor refers to a few of the bits and the sum ranges over all consistent assignments [30]. In such a sum the local factors are individually simple, and the global object, the full assignment, is what carries the cost.

The analogy with the distinction between local admissibility and global compatibility in distributed systems is natural, and it will be developed in Part XIII. It should be stated here as no more than an analogy. The definition of VNP gives a precise algebraic sense to the notion of efficiently computable local contributions with an inefficient aggregate, but it does not by itself supply a general theory of coordination, and the extension to systems outside algebraic complexity requires separate definitions and arguments.

## 4.4 Algebraic Completeness

The notion of completeness allows a statement about an entire class to be reduced to a statement about one family. A polynomial $f$ is a projection of a polynomial $g$ if $f$ is obtained from $g$ by substituting for each variable of $g$ either a variable or a constant of the field. A family $(f_n)$ is a polynomially bounded projection of $(g_n)$ if $f_n$ is a projection of $g_{t(n)}$ for a polynomially bounded $t$. Since substitution does not increase circuit size, the class VP is closed under such projections, and a family is called VNP-complete if it lies in VNP and every family in VNP is a polynomially bounded projection of it. If a VNP-complete family were in VP, then all of VNP would be in VP, and conversely a superpolynomial lower bound for one complete family implies that the class does not collapse.

Valiant proved that the permanent is VNP-complete over every field of characteristic

other than two [34], and the source manuscript records that the Hamiltonian cycle polynomial is VNP-complete over every field [30]. The lower bound for a complete family therefore establishes a limitation for the class as a whole, which is why an argument about the permanent and the Hamiltonian cycle polynomial is sufficient to separate VP from VNP. The reduction in the opposite direction, from a clause sum to the permanent, is used in the manuscript in an explicit form, and it is the point at which the determinant escapes, as Chapter 23 will discuss.

## 4.5 The Separation Problem

Valiant's conjecture is the statement

$$\mathrm{VP} \subsetneq \mathrm{VNP}.$$

Its equivalent formulation in terms of the permanent asserts that the family of permanents of $m \times m$ matrices has no circuits of size polynomial in $m$, and the same holds for the Hamiltonian cycle polynomial. A proof that a complete family requires superpolynomial circuits establishes the conjecture, and such a proof is a considerably weaker achievement than a determination of the true complexity of the family. The best known upper bounds for the permanent are exponential in $m$, of the form $2^{O(m)}$, obtained from Ryser's formula and its refinements [29, 14], and a lower bound of the form $2^{\Omega((\log m)^{4/3})}$ leaves a very large gap between what is known to be necessary and what is known to be sufficient. To prove that a family cannot be computed in polynomial size is to establish a fact about the boundary of a class, and it does not determine the location of the family within the space beyond that boundary. The distinction between superpolynomial hardness and exact complexity will recur whenever a particular lower bound is assessed.

## 4.6 Formal Statements

Write $n^{O(1)}$ for a quantity bounded by $n^a + a$ for some constant $a$.

**Definition 4.1** (VP)**.** $\mathrm{VP}_{\mathbb{F}}$ is the set of families $(f_n)$ of polynomials over $\mathbb{F}$ with $\mathrm{vars}(f_n) = n^{O(1)}$, $\deg f_n = n^{O(1)}$, and $L_{\mathbb{F}}(f_n) = n^{O(1)}$.

**Definition 4.2** (VNP)**.** $\mathrm{VNP}_{\mathbb{F}}$ is the set of families $(f_n)$ with $f_n(x) = \sum_{e \in \{0,1\}^{r(n)}} g_n(x, e)$ for some $(g_n) \in \mathrm{VP}_{\mathbb{F}}$, where $r(n) = n^{O(1)}$.

**Proposition 4.3.** $\mathrm{VP}_{\mathbb{F}} \subseteq \mathrm{VNP}_{\mathbb{F}}$.

*Proof.* Take $r(n) = 0$, so that the sum has a single term. □

**Definition 4.4** (Polynomial Projection)**.** Write $f \leq_{\mathrm{proj}} g$ if $f$ is obtained from $g$ by substituting for each variable of $g$ a variable or a constant of $\mathbb{F}$. A family $(f_n)$ is a polynomially bounded projection of $(h_n)$ if $f_n \leq_{\mathrm{proj}} h_{t(n)}$ for some $t(n) = n^{O(1)}$.

**Proposition 4.5** (Projection Monotonicity)**.** *If $f \leq_{\text{proj}} g$, then $L_{\mathbb{F}}(f) \leq L_{\mathbb{F}}(g)$.*

*Proof.* Apply the substitution to the input gates of a circuit for $g$: each variable gate becomes a variable gate or a constant gate, so the size is unchanged, and the gate for $g$ now computes $f$. □

**Theorem 4.6** (Completeness Principle)**.** *Let $(h_n)$ be* VNP*-complete under polynomially bounded projections, that is, $(h_n) \in \mathrm{VNP}_{\mathbb{F}}$ and every family in $\mathrm{VNP}_{\mathbb{F}}$ is a polynomially bounded projection of it. Then $\mathrm{VP}_{\mathbb{F}} \neq \mathrm{VNP}_{\mathbb{F}}$ if and only if $(h_n) \notin \mathrm{VP}_{\mathbb{F}}$.*

*Proof.* If $(h_n) \notin \mathrm{VP}_{\mathbb{F}}$, then $(h_n) \in \mathrm{VNP}_{\mathbb{F}} \setminus \mathrm{VP}_{\mathbb{F}}$, and $\mathrm{VP}_{\mathbb{F}} \subsetneq \mathrm{VNP}_{\mathbb{F}}$ because the inclusion holds. Conversely suppose $(h_n) \in \mathrm{VP}_{\mathbb{F}}$ and let $(f_n) \in \mathrm{VNP}_{\mathbb{F}}$. Then $f_n \leq_{\text{proj}} h_{t(n)}$ with $t$ polynomially bounded, so $L_{\mathbb{F}}(f_n) \leq L_{\mathbb{F}}(h_{t(n)}) = n^{O(1)}$ by projection monotonicity, and the degree and number of variables of $f_n$ are polynomially bounded because $(f_n) \in \mathrm{VNP}_{\mathbb{F}}$. Hence $(f_n) \in \mathrm{VP}_{\mathbb{F}}$ and the classes coincide. □

The relevant completeness theorems, that the permanent is complete over fields of characteristic other than two and the Hamiltonian cycle polynomial over every field, are inherited from Valiant's framework [34] and from the manuscript's explicit constructions.

# Chapter 5

# The Determinant, Permanent, and Hamiltonian Cycle Polynomial

## 5.1 The Determinant

The determinant of an $m \times m$ matrix $X = (x_{i,j})$ of variables is given by the Leibniz expansion

$$\det(X) = \sum_{\sigma \in S_m} \operatorname{sgn}(\sigma) \prod_{i=1}^{m} x_{i,\sigma(i)},$$

a sum of $m!$ monomials with signs. The expanded expression is enormous, and nevertheless the determinant has arithmetic circuits of size polynomial in $m$. Berkowitz gave a division-free algorithm using only additions, subtractions, and multiplications [7], and the existence of such an algorithm shows that the signs in the expansion produce cancellations and linear-algebraic structure sufficient to organize the computation without enumerating the terms. Among the properties responsible is the multiplicativity of the determinant under matrix products and its behavior under elementary row operations, which allow it to be evaluated by a procedure resembling elimination. The determinant therefore exhibits a polynomial with factorially many monomials that nevertheless lies in VP.

## 5.2 The Permanent

The permanent is defined by the same expansion with the signs removed,

$$\operatorname{perm}(X) = \sum_{\sigma \in S_m} \prod_{i=1}^{m} x_{i,\sigma(i)}.$$

The two polynomials contain the same monomials, and they differ only in the coefficients, which are all equal to one in the permanent and are the signs of the permutations in the determinant. The removal of the sign structure eliminates the cancellations and the algebraic

identities on which the efficient computation of the determinant rests, and the known complexity of the two changes sharply. This describes the available constructions and does not amount to a proof of hardness: the efficient determinant circuits exploit identities that the permanent does not generally satisfy, and the absence of determinant-like cancellation is not by itself a lower bound. Evaluating the permanent of a matrix with entries in $\{0, 1\}$ is complete for the counting class #P [35], although that is a statement about Boolean computation and not about arithmetic circuits for the formal polynomial. The best known arithmetic circuits for the permanent are of exponential size.

The observation that the two polynomials contain the same monomials has an important methodological consequence. A lower bound for the permanent cannot be obtained merely by counting monomials, because the determinant has the same monomials and has small circuits. Compact circuits are able to represent polynomials with exponentially many monomials: the product $\prod_{i=1}^{n}(1 + x_i)$ has $2^n$ monomials and a circuit of linear size. Any valid lower-bound argument must therefore exploit the way in which the monomials are organized and combined, and not merely their number.

## 5.3 The Hamiltonian Cycle Polynomial

The Hamiltonian cycle polynomial is

$$\mathrm{HC}_m(X) = \sum_{\substack{\sigma \in S_m \\ \sigma \text{ is a single cycle}}} \prod_{i=1}^{m} x_{i,\sigma(i)},$$

where the sum ranges over the $(m-1)!$ permutations whose cycle decomposition consists of one cycle through all $m$ points. When the variables are replaced by the entries of the adjacency matrix of a directed graph, the polynomial counts the directed Hamiltonian cycles of the graph, and in general it is a weighted count of such cycles.

Each term of the sum is a choice of a successor $\sigma(i)$ for every index $i$. Locally, such a choice is unconstrained except that distinct indices must receive distinct successors, which makes $\sigma$ a permutation. The additional requirement that the permutation form a single cycle is a global condition, for it cannot be verified by inspecting the choices at any bounded number of indices. The polynomial is thus a weighted aggregation over globally compatible cyclic configurations, and this makes it a natural member of the family of examples in which local choices are easily described and the global constraint carries the difficulty.

## 5.4 Characteristic Two

In characteristic two the integer $-1$ equals 1, and hence every sign $\mathrm{sgn}(\sigma) \in \{1, -1\}$ is equal to 1. The Leibniz expansion of the determinant then coincides term by term with the

expansion of the permanent, so that

$$\operatorname{perm}(X) = \det(X)$$

over every field of characteristic two. The determinant has circuits of polynomial size over every field, and the permanent therefore has polynomial-size circuits in characteristic two. This is why no superpolynomial lower bound for the permanent can hold in characteristic two, and why the statement of the lower bound of the source manuscript excludes the permanent in that characteristic [30]. The permanent cannot serve as the universal obstruction over all fields. The Hamiltonian cycle polynomial supplies a complete family without this degeneracy, since the reduction to it requires no division and holds over every field, and this is the reason it appears alongside the permanent in the statement of the theorem.

## 5.5 Why Superficially Similar Expressions Can Differ Computationally

The determinant and the permanent together establish that computational difficulty cannot be inferred from expression length, from the number of configurations represented, or from the superficial similarity of two definitions. Both polynomials are sums over the same $m!$ permutations, both have the same monomials, and their definitions differ by a sign. The one is computable in polynomial size and the other, on the evidence of the lower bounds under discussion, is not. The Hamiltonian cycle polynomial is a sum over a subset of the same index set and is complete for the class that the permanent is complete for, so that restricting the sum to a smaller and more structured set of permutations does not make the computation easier.

What distinguishes the cases is the organization of algebraic dependencies. The signs of the determinant produce multiplicative structure that the polynomial inherits from matrix algebra, and the permanent has no counterpart of this structure. A lower-bound argument that applies to the permanent and to the Hamiltonian cycle polynomial but not to the determinant must therefore detect a property of the permanent that depends on the absence of the cancellations of the determinant. This requirement is a constraint on every proof of the separation, and it provides a diagnostic that will be applied to the manuscript's argument in Chapter 23.

## 5.6 Formal Statements

**Definition 5.1.** For an $m \times m$ matrix $X = (x_{i,j})$ of variables let

$$\det_m(X) = \sum_{\sigma \in S_m} \operatorname{sgn}(\sigma) \prod_{i=1}^{m} x_{i,\sigma(i)}, \qquad \operatorname{perm}_m(X) = \sum_{\sigma \in S_m} \prod_{i=1}^{m} x_{i,\sigma(i)}.$$

Let $S_m^{\mathrm{cyc}} \subseteq S_m$ be the set of permutations consisting of a single cycle through all $m$ points, and

$$\mathrm{HC}_m(X) = \sum_{\sigma \in S_m^{\mathrm{cyc}}} \prod_{i=1}^{m} x_{i,\sigma(i)}.$$

**Proposition 5.2** (Monomial Cardinalities)**.** $|\operatorname{supp}(\det_m)| = m!$, $|\operatorname{supp}(\operatorname{perm}_m)| = m!$, *and* $|\operatorname{supp}(\mathrm{HC}_m)| = (m-1)!$.

*Proof.* The monomial $\prod_i x_{i,\sigma(i)}$ determines $\sigma$, so distinct permutations give distinct monomials and no two terms combine. Every coefficient is $\pm 1$, which is nonzero in every field, so every monomial lies in the support. There are $m!$ permutations and $(m-1)!$ of them are single cycles. □

**Proposition 5.3** (Characteristic-Two Degeneracy)**.** *If* $\operatorname{char}(\mathbb{F}) = 2$*, then* $\det_m = \operatorname{perm}_m$ *over* $\mathbb{F}$*.*

*Proof.* Every sign $\operatorname{sgn}(\sigma) \in \{1, -1\}$ equals 1 in $\mathbb{F}$ because $-1 = 1$. □

**Proposition 5.4** (Monomial Abundance Is Not a Lower Bound)**.** *Let* $f_n(x, y) = \prod_{i=1}^{n}(x_i + y_i)$*. Then* $|\operatorname{supp}(f_n)| = 2^n$ *while* $L_{\mathbb{F}}(f_n) \leq 4n - 1$*.*

*Proof.* Expanding the product, each monomial chooses $x_i$ or $y_i$ for every $i$, and these $2^n$ monomials are distinct and have coefficient one. A circuit has $2n$ variable gates, $n$ sum gates, and $n - 1$ product gates. □

Thus exponentially many monomials do not imply superpolynomial arithmetic circuit complexity.

**Inherited Result 5.5** (Determinant in VP)**.** $(\det_m) \in \mathrm{VP}$*. This is due to Berkowitz's division-free algorithm [7].*

The contrast between the determinant and the permanent therefore cannot be explained by monomial cardinality alone.

# Part III

# The Claimed Superpolynomial Separation

# Chapter 6

# Statement, Scope, and Interpretation of the Result

## 6.1 The Principal Lower Bound

The source manuscript, titled *VP ≠ VNP over Every Field* [30], reports the following lower bound. The statement is reproduced as an inherited result, to be reconstructed and assessed in the chapters that follow.

**Inherited Result 6.1** (Principal Lower Bound)**.** *There is an absolute constant $c > 0$ such that for every $p$ that is zero or prime there is a threshold $m_0$ with the following property. Over every field of characteristic $p$ and for every $m \geq m_0$, every arithmetic circuit for* $\mathrm{HC}_m$ *has more than*

$$m^{c(\log_2 m)^{1/3}}$$

*gates, and so does every circuit for* $\mathrm{perm}_m$ *when $p \neq 2$.*

Several features of the statement should be read carefully. The constant $c$ is absolute, in the sense that it does not depend on the field or on its characteristic, whereas the threshold $m_0$ is allowed to depend on the characteristic $p$. The manuscript does not claim that the exponent $\frac{1}{3}$ is optimal. In characteristic two the permanent coincides with the determinant, and the statement excludes it for that reason. Since the families $(\mathrm{perm}_m)$ and $(\mathrm{HC}_m)$ lie in VNP, the reported bound yields the corollary that VP is a proper subset of VNP over every field. The manuscript states that the proofs are contained in an associated Lean development and in its appendix, and the reliability of that claim is a separate question, examined in Part X.

## 6.2 Asymptotic Interpretation

The bound can be rewritten by using $m = 2^{\log_2 m}$, which gives

$$m^{c(\log_2 m)^{1/3}} = 2^{c(\log_2 m)^{4/3}}.$$

The lower bound is therefore of the form $2^{\Omega((\log m)^{4/3})}$. For every fixed exponent $a$ the ratio

$$\frac{m^{c(\log_2 m)^{1/3}}}{m^a} = 2^{\left(c(\log_2 m)^{1/3} - a\right)\log_2 m} \longrightarrow \infty$$

as $m \to \infty$, since the exponent becomes positive and unbounded once $(\log_2 m)^{1/3}$ exceeds $a/c$. The bound eventually exceeds every fixed polynomial, and this is what makes it a separation.

The bound is quasipolynomial rather than exponential. A function of the form $2^{(\log m)^{O(1)}}$ is far smaller than $2^{m^\delta}$ for any $\delta > 0$, and the upper bounds known for the permanent are of the form $2^{O(m)}$ [29, 14]. The manuscript explicitly recognizes the remaining gap between the lower bound and the exponential upper bounds [30]. It also reports that in one of the two cases into which its argument divides, the bound improves to $2^{c(\log m)^2/\log\log m}$, with constants that depend on an integer that the proof does not determine. Figure 6.1 compares three growth rates on a logarithmic scale for the circuit size, with the lower-bound constant normalized to one.

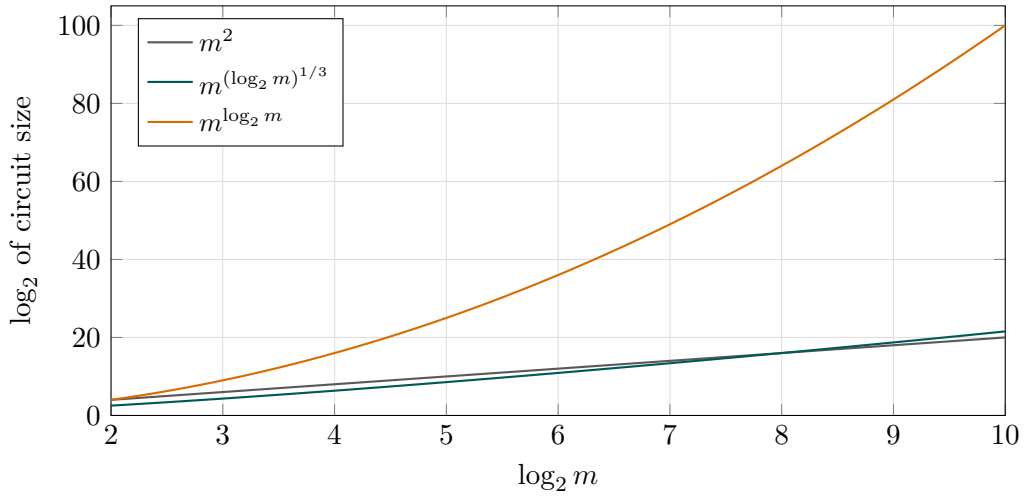


Figure 6.1: Illustrative values of $\log_2$ circuit size against $\log_2 m$, with the lower-bound constant normalized to one. These are growth functions chosen for comparison, not measured circuit sizes or explicit numerical bounds from the source manuscript.

The figure shows that over the plotted range the quasipolynomial lower bound with the exponent $\frac{1}{3}$ remains close to a quadratic, which indicates how slowly the bound escapes polynomial growth at moderate sizes, while the function $m^{\log_2 m}$ separates sharply. The comparison is intended only to display the shapes of the three functions.

## 6.3 What the Theorem Does Not Establish

The reported result does not determine the optimal arithmetic circuit complexity of the permanent, since the exponential upper bounds and the quasipolynomial lower bound remain far apart. It does not provide a practical threshold for computational difficulty: the threshold $m_0$ is given by an existence statement, and in the case of the manuscript's argument that depends on an integer, the proof supplies no value for it, so that the theorem does not specify any particular size at which the stated bound applies. It does not establish exponential hardness, and the manuscript says so. Finally, it does not resolve the Boolean problem P versus NP. An arithmetic circuit must compute the polynomial for all inputs in the field and its extensions, and not merely for the inputs 0 and 1, and the lower bound does not apply to Boolean circuits [30]. These limitations are consequences of the logic of the statement and not shortcomings of its presentation, and they are the starting point for the analysis of the boundaries of the theorem in Part VIII.

## 6.4 Exact Versus Approximate Computation

The manuscript extends its bound to a notion of approximate computation drawn from border complexity. Let $k$ be a field and let $k((\varepsilon))$ be the field of formal Laurent series in a variable $\varepsilon$ with coefficients in $k$. A circuit over $k((\varepsilon))$ approximates a polynomial $f$ over $k$ if it computes a polynomial of the form

$$f + \varepsilon g,$$

where all coefficients of $g$ lie in the power series ring $k[[\varepsilon]]$. The class $\overline{\mathrm{VP}}$ is defined like VP with approximating circuits in place of circuits.

Over the real numbers this notion captures limits. If the constants of a circuit are polynomials in $\varepsilon$ and $1/\varepsilon$, then substituting a small positive real number for $\varepsilon$ gives a circuit over $\mathbb{R}$ of the same size computing $f+\varepsilon g$, and as $\varepsilon \to 0$ these polynomials tend to $f$ coefficient by coefficient. The polynomial $f$ is a limit of polynomials having small circuits, although it need not have a small circuit itself, because constants such as $1/\varepsilon$ become unbounded in the limit. The definition by formal series expresses the same idea over an arbitrary field.

A lower bound for approximating circuits is stronger than a lower bound restricted to exact computation. Every exact circuit over $k$ is also a circuit over $k((\varepsilon))$ computing $f$ with $g = 0$, so that the class of approximating circuits contains the class of exact circuits, and a lower bound against the larger class implies the lower bound against the smaller. The manuscript claims that its bound holds for approximating circuits and that, consequently,

$$\mathrm{VNP} \not\subseteq \overline{\mathrm{VP}}$$

over every field [30]. It further draws from this a consequence for the determinantal complexity of the permanent, which it identifies as the one numbered statement of the paper

that is not formalized in Lean, a qualification that will be taken up in Chapters 24 and 27.

## 6.5 Historical Context and Prior Lower Bounds

The significance of the reported bound is best judged against the state of lower bounds for arithmetic circuits. For unrestricted circuits, that is, circuits with no constraint on depth or structure, the lower bounds that had been obtained were remarkably weak. The best previous lower bound for an explicit polynomial of degree $d$ in $n$ variables, as the manuscript records, is of order $n \log d$, due to Strassen and to Baur and Strassen [31, 6]. For the permanent of an $m \times m$ matrix, which has $m^2$ variables and degree $m$, this is of order $m^2 \log m$, and no lower bound of larger order was known. A superpolynomial bound for general circuits would accordingly be a very large advance on that state of knowledge.

Much stronger bounds had been established for restricted models. Exponential lower bounds are known for monotone circuits [20] and, more recently, a separation of monotone VP from monotone VNP was obtained [37]. Exponential bounds hold for noncommutative formulas and algebraic branching programs [25]. Superpolynomial bounds are known for multilinear formulas [28] and, by the work of Limaye, Srinivasan, and Tavenas, for circuits of every constant depth in characteristic zero or large characteristic [22], a result that was subsequently extended to all fields [13]. For formulas and algebraic branching programs the known bounds are only polynomial. The programme of depth reduction, which began with Hyafil and with Valiant, Skyum, Berkowitz, and Rackoff [19, 36] and continued through the reductions to depth four and depth three [1, 21, 33, 15], connects these restricted models to general circuits and explains why lower bounds for bounded depth were regarded as a route toward the general case. The geometric complexity theory programme of Mulmuley and Sohoni [23, 24] approaches the border version of the conjecture through representation theory, and the manuscript emphasizes that its own argument uses neither representation theory nor orbit closures.

Further results of 2026 concern different computational models and are not to be read as a sequence of improvements on one another. Raz obtained a lower bound of order $n^{3/2}$ on the number of multiplication gates of noncommutative arithmetic circuits computing an explicit polynomial of degree $n$ in $n$ variables over every field, and more generally of order $d\sqrt{n}$ for degree $d$ [27]. An OpenAI manuscript of September 2026, accompanied by a Lean development, claims that the border determinantal complexity of the permanent over the complex numbers is $\Omega(m^3)$ [26]. The explicit bound covered by its formalization is $m^3/(5\,529\,600\,e)$ for $m \geq 1408$, and the manuscript's further consequences for algebraic branching programs lie outside the stated scope of that formalization. It is cited as a research preprint. Guo derives from that manuscript lower bounds for permanent circuits with valid division in characteristic zero, in a model that counts nonscalar multiplications and divisions and treats additions and scalar operations as free [17]. The unconditional result is that the circuit size is at least $\frac{1}{12} n^2 \log_2 \log_2 n$ in the limit inferior. A second bound of order $n^2 \log n$ combines twelve machine-checked declarations from the OpenAI development with

additional written arguments, and a third is explicitly conditional on a proposition of the OpenAI manuscript. The three works are cited under the restrictions of their own models, and none of them is a bound for the general arithmetic circuits of the manuscript under study.

The historical account suggests why the claimed result, if correct, would represent a substantial technical advance: it concerns unrestricted circuits, with no constraint on depth, on constants, or on cancellations, and its bound is superpolynomial where the previous record was nearly linear in the number of variables. The criterion introduced in Chapter 2 prevents this from being equated with conceptual novelty. A large improvement in a numerical bound is evidence of difficulty overcome, and whether it reflects a reusable structure is a question about the method, which the manuscript's introduction describes as a passage from circuits to tensors, from tensors to metric geometry, and from curvature to a dichotomy between two cases. The next parts reconstruct that passage.

## 6.6 Formal Statements

Inherited Result 6.1 has the quantifier structure

$$\begin{aligned}&\exists c > 0,\ \forall p \in \{0\} \cup \{\text{primes}\},\ \exists m_0 \in \mathbb{N},\\ &\forall \mathbb{F} : \operatorname{char}(\mathbb{F}) = p,\ \forall m \geq m_0 :\\ &\qquad L_{\mathbb{F}}(\mathrm{HC}_m) > m^{c(\log m)^{1/3}},\end{aligned}$$

and the corresponding assertion for $\mathrm{perm}_m$ when $p \neq 2$. The order of the quantifiers is important: the constant $c$ is absolute, whereas the threshold may depend on the characteristic.

**Proposition 6.2** (Superpolynomial Growth)**.** *For every fixed $a$ and $c > 0$,*

$$\lim_{m \to \infty} \frac{m^{c(\log m)^{1/3}}}{m^a} = \infty.$$

*Proof.* $\log\bigl(m^{c(\log m)^{1/3}}/m^a\bigr) = c(\log m)^{4/3} - a \log m = \log m\bigl(c(\log m)^{1/3} - a\bigr)$, which tends to infinity because the second factor does. □

**Corollary 6.3.** *Conditional on Inherited Result 6.1,* $\mathrm{VP}_{\mathbb{F}} \subsetneq \mathrm{VNP}_{\mathbb{F}}$ *over every field* $\mathbb{F}$*.*

*Proof.* Fix $\mathbb{F}$ and let $p$ be its characteristic. The family $(\mathrm{HC}_m)$ has $m^2$ variables and degree $m$ and belongs to $\mathrm{VNP}_{\mathbb{F}}$. If it belonged to $\mathrm{VP}_{\mathbb{F}}$ its circuit size would be at most $m^a + a$ for some $a$ and all $m$, which contradicts the lower bound for large $m$ by the previous proposition. By Theorem 4.6 or directly, $\mathrm{VP}_{\mathbb{F}} \subsetneq \mathrm{VNP}_{\mathbb{F}}$. □

**Proposition 6.4** (Growth Hierarchy)**.** *For fixed $a$, $b > 0$ and $c > 0$, $m^a = o\bigl(2^{c(\log m)^{4/3}}\bigr)$ and $2^{c(\log m)^{4/3}} = o(2^{bm})$.*

*Proof.* The first claim is the previous proposition with $m^{c(\log m)^{1/3}} = 2^{c(\log m)^{4/3}}$. For the second, $c(\log m)^{4/3} - bm \to -\infty$ because $(\log m)^{4/3} = o(m)$. □

The claimed bound therefore separates polynomial from quasipolynomial complexity without establishing exponential complexity.

Throughout the monograph the distinction between a reported separation and an independently reconstructed separation is maintained. Statements that the argument of the source is logically complete are attributions to [30] and are not claims that the present reconstruction is complete, and the reconstruction is not complete in this version.

**Proof Obligation 6.5** (Principal Proof Obligation)**.** *Chapters 7 to 24 must establish Inherited Result 6.1 and not merely invoke it.*

# Part IV

# From Algebraic Complexity to Metric Geometry

# Chapter 7

# Tensor Restrictions

## 7.1 Multilinear Representations

Let $V$ be a finite set whose elements serve only as labels, and call them parties. For each party $v$ fix a finite-dimensional vector space $E_v = \mathbb{F}^{n_v}$ with $n_v \geq 1$. A $V$-tensor is an element of the tensor product

$$\mathcal{T}_V(\mathbf{n}) = \bigotimes_{v \in V} \mathbb{F}^{n_v},$$

that is, an array over $\mathbb{F}$ with one index for each party. The term party is an index label and carries no implication of agency or individuality. The interest of the construction lies in what a tensor represents: a table of joint dependencies among components that can be described one at a time but whose combined behavior is not determined by those separate descriptions. A product tensor $a_1 \otimes \cdots \otimes a_n$ records no dependence at all, since each component contributes independently, whereas a tensor that is not a product records a correlation that no list of component-wise data reproduces.

## 7.2 Tensor Products and Independent Composition

Two tensors on the same set of parties are composed partywise. If $s \in \bigotimes_v \mathbb{F}^{m_v}$ and $t \in \bigotimes_v \mathbb{F}^{n_v}$, their product $st$ lies in $\bigotimes_v \mathbb{F}^{m_v n_v}$, each party holding the pair of its indices in $s$ and in $t$ [30]. The product represents the simultaneous organization of two dependency structures on the same parties, with the combined index at each party. It is commutative and associative up to relabeling of indices, and its unit is the tensor $\mathbf{1}$ in which every party has a one-element index set and the entry is 1.

The behavior of an invariant under repeated products is the central question of the asymptotic theory. The rank of a matrix is multiplicative under the Kronecker product, so that matrix rank across a partition of the parties is exactly multiplicative. Tensor rank is in general only submultiplicative. The three-party tensor commonly denoted $W$ has tensor rank 3 and border rank 2, and the rank of its $n$-fold power grows like $2^{n+o(n)}$, so that its asymptotic rank is strictly smaller than its rank. Properties of a tensor therefore divide into

those preserved exactly by products and those that emerge only at larger scales, and the spectral functionals of the next chapter are designed to capture the first kind.

## 7.3 Restrictions and Local Transformations

A restriction transforms a tensor by linear maps that act on each party separately. If $A_v : \mathbb{F}^{n_v} \to \mathbb{F}^{m_v}$ are linear maps, then

$$s = \Big(\bigotimes_{v \in V} A_v\Big) t$$

is a restriction of $t$. Each map $A_v$ sees only the index of its own party, so that the transformation is local in the sense that no party has access to another party's index. The central question of the theory is whether a complicated global dependency structure can be generated from a simpler one by such local transformations, and the answer is expressed by the preorder introduced next.

## 7.4 Restriction as a Resource Preorder

Write $s \preceq t$ when $s$ is a restriction of $t$. The relation is reflexive and transitive, so that it is a preorder, and mutual restriction is an equivalence relation whose classes can be identified. The preorder is a mathematical account of relative computational resources: if $s \preceq t$, then $t$ can serve wherever $s$ is needed, and any resource sufficient to produce $t$ is sufficient to produce $s$. The account is relational and requires no scalar measure imposed from outside. A scalar enters only through functionals that respect the preorder, and these are the subject of Chapter 8. In the manuscript the equivalence classes are identified, so that the product is commutative and associative with unit $\mathbf{1}$, every nonzero tensor has $\mathbf{1}$ as a restriction, and every tensor is covered by finitely many copies of $\mathbf{1}$ [30].

## 7.5 The Insufficiency of Individual Component Descriptions

The complexity of a tensor may reside wholly in the relationships among its components and not in the dimensions or descriptions of the components. Two matrices of size $n \times n$, regarded as tensors on two parties, have the same index dimensions at both parties. The identity matrix has rank $n$, and a matrix with a single nonzero entry has rank 1. Neither is a restriction of the other in both directions, because mutual restriction of matrices holds exactly when the ranks agree. The two parties see identical individual data in both cases, and the difference lies entirely in the joint structure. This is the algebraic form of a familiar situation in which locally consistent records fail to determine a globally admissible structure, and it motivates the later discussion of gluing in Part XIII.

## 7.6 Formal Statements

**Definition 7.1** (Tensor Space)**.** For a finite index set $V$ and a vector $\mathbf{n} = (n_v)_{v \in V}$ of positive integers, let $\mathcal{T}_V(\mathbf{n}) = \bigotimes_{v \in V} \mathbb{F}^{n_v}$. The collection of these spaces over all dimension vectors is denoted $\mathcal{T}_V$.

**Definition 7.2** (Local Restriction)**.** For $s \in \mathcal{T}_V(\mathbf{m})$ and $t \in \mathcal{T}_V(\mathbf{n})$ write $s \preceq t$ if there are linear maps $A_v : \mathbb{F}^{n_v} \to \mathbb{F}^{m_v}$ with $s = (\bigotimes_v A_v)t$.

**Proposition 7.3** (Preorder Structure)**.** *The relation $\preceq$ is reflexive and transitive: $t \preceq t$, and $r \preceq s$ and $s \preceq t$ imply $r \preceq t$.*

*Proof.* Reflexivity follows by taking every $A_v$ to be an identity map. If $r = (\bigotimes_v B_v)s$ and $s = (\bigotimes_v A_v)t$, then $r = (\bigotimes_v B_v)(\bigotimes_v A_v)t = (\bigotimes_v B_v A_v)t$, because the tensor product of linear maps is functorial. □

**Proposition 7.4** (Compositional Monotonicity)**.** *If $s_1 \preceq t_1$ and $s_2 \preceq t_2$, then $s_1 s_2 \preceq t_1 t_2$, where the product is the partywise tensor product.*

*Proof.* Let $s_i = (\bigotimes_v A_v^{(i)})t_i$. By the mixed-product property of the Kronecker product, applied at each party, $s_1 s_2 = (\bigotimes_v (A_v^{(1)} \otimes A_v^{(2)}))(t_1 t_2)$. □

**Definition 7.5** (Restriction Equivalence)**.** Write $s \simeq t$ if $s \preceq t$ and $t \preceq s$. The quotient by $\simeq$ is a partially ordered resource structure.

*Remark* 7.6. A tensor that is not a product records a joint dependence among its components. The word correlation is used informally and carries no probabilistic meaning, since tensors over arbitrary fields need not be distributions and their entries need not admit interpretation as frequencies or amplitudes.

# Chapter 8

# Spectral Points and Multiplicative Invariants

## 8.1 The Asymptotic Spectrum

Strassen's asymptotic spectrum is the set of functionals on tensors that are monotone under restriction, multiplicative under the product, additive under direct sums, and normalized on unit tensors [32]. The motivation is that the cost of a tensor under repeated composition is better captured by a family of such functionals than by a single rank measure, since each functional isolates a different aspect of how the tensor behaves when many copies are combined. Rank itself is submultiplicative and not multiplicative, and for this reason it gives only upper estimates of asymptotic cost.

The source manuscript works with a larger class that it calls spectral points, defined by weaker axioms, and remarks that its class is larger than Strassen's, since its points need not be additive under direct sums or normalized on the diagonal unit tensors $\langle r \rangle = \sum_{i=1}^{r} e_i^{(1)} \otimes \cdots \otimes e_i^{(k)}$, for which the standard spectrum requires $\varphi(\langle r \rangle) = r$ [30]. Normalization at the multiplicative identity $\mathbf{1}$ is retained, and the two normalizations should not be confused. The enlargement is harmless for the purposes of obstruction, because the argument needs only that spectral points exist in sufficient variety and that they are monotone and multiplicative.

## 8.2 The Defining Properties of Spectral Points

A spectral point is a function $\varphi \geq 0$ on the tensors on a given set of parties satisfying three conditions. It is normalized, $\varphi(\mathbf{1}) = 1$. It is multiplicative, $\varphi(st) = \varphi(s)\varphi(t)$. And it is subadditive on covers: if $s = \sum_j s_j$ with each $s_j$ a restriction of $t_j$, then

$$\varphi(s) \leq \sum_j \varphi(t_j).$$

One cover consisting of a single tensor shows that $\varphi$ is monotone under restriction, and the empty cover shows that $\varphi(0) = 0$. Since $\mathbf{1}$ is a restriction of every nonzero tensor, $\varphi(t) \geq 1$ for every $t \neq 0$. These are the functional properties from which the subsequent geometry is constructed.

## 8.3 Why Multiplicativity Matters

Multiplicativity governs repeated composition. If $\varphi(t^n) = \varphi(t)^n$, then

$$\log \varphi(t^n) = n \log \varphi(t),$$

and the logarithm converts the multiplicative composition of tensors into an additive quantity. This is what makes it possible to compare tensors through exponential growth rates and to treat networks of wires, whose values are products, by sums of lengths, as Chapter 9 will show. It is also the property that makes the existence of spectral points a delicate matter, since a function that is only submultiplicative, such as the rank, is easy to construct and a function that is exactly multiplicative is not.

## 8.4 Nonexplicit Spectral Points

The distinction between proving that a functional exists and exhibiting an algorithm that evaluates it is sharp in this theory. Submultiplicative functions arise naturally as infima over covers, whereas exact multiplicativity does not. The manuscript obtains spectral points below a given upper functional by an application of Zorn's lemma, taking a minimal upper functional and showing that minimality forces multiplicativity [30]. The spectral points that its argument uses are consequently not explicit. Few points of the asymptotic spectrum are known explicitly, namely the ranks of flattenings, the support functionals on oblique tensors, and, over the complex numbers, the quantum functionals, and the manuscript states that the points it uses are not among them. This distinction will become important in the discussion of constructive knowledge and operational accessibility in Part IX.

## 8.5 A Plurality of Obstruction Measures

Different spectral points may detect different forms of computational obstruction, and no single point is privileged. The rank of a flattening across a fixed partition is one example, and the function that is 1 on all nonzero tensors is another. New points can be generated from old: if $\varphi$ is a spectral point and $0 < a \leq 1$, then $\varphi^a$ is again a spectral point. The obstruction is therefore better represented by a family of admissible valuations than by a universally privileged scalar. This observation resembles a point made in other settings, where a single numerical score cannot capture a structured notion of adequacy and a family of admissible measures is required. The resemblance is an analogy and not an established equivalence, and the mathematical content of the present chapter does not depend on it.

## 8.6 Formal Statements

**Definition 8.1** (Spectral Point)**.** A spectral point on $V$ is a function $\varphi : \mathcal{T}_V \to \mathbb{R}_{\geq 0}$ such that

$$\varphi(\mathbf{1}) = 1, \qquad \varphi(st) = \varphi(s)\varphi(t),$$

and such that $s = \sum_{j=1}^{N} s_j$ with $s_j \preceq t_j$ implies $\varphi(s) \leq \sum_{j=1}^{N} \varphi(t_j)$.

**Proposition 8.2** (Restriction Monotonicity)**.** *If $s \preceq t$, then $\varphi(s) \leq \varphi(t)$. Moreover $\varphi(0) = 0$ and $\varphi(t) \geq 1$ for every nonzero tensor $t$.*

*Proof.* Take $N = 1$, $s_1 = s$ and $t_1 = t$ in the cover axiom. For $N = 0$ the empty sum is the zero tensor, which gives $\varphi(0) \leq 0$. A nonzero tensor has $\mathbf{1}$ as a restriction, so $1 = \varphi(\mathbf{1}) \leq \varphi(t)$. □

**Proposition 8.3** (Tensor-Power Scaling)**.** *For every tensor $t$ and every integer $n \geq 1$, $\varphi(t^n) = \varphi(t)^n$, and for $t \neq 0$, $\log \varphi(t^n) = n \log \varphi(t)$.*

*Proof.* By induction on $n$ using multiplicativity. The logarithm is defined because $\varphi(t) \geq 1 > 0$ for nonzero $t$. □

**Proposition 8.4** (Spectral Obstruction)**.** *If $\varphi(s) > \varphi(t)$ for some spectral point $\varphi$, then $s \not\preceq t$.*

*Proof.* If $s \preceq t$, then $\varphi(s) \leq \varphi(t)$ by monotonicity. □

**Proposition 8.5** (Powers of Spectral Points)**.** *If $\varphi$ is a spectral point and $0 < a \leq 1$, then $\varphi^a$ is a spectral point.*

*Proof.* Normalization and multiplicativity are preserved by the power. For a cover, $\varphi(s)^a \leq \left(\sum_j \varphi(t_j)\right)^a \leq \sum_j \varphi(t_j)^a$, where the last step uses the subadditivity of $x \mapsto x^a$ on $[0, \infty)$ for $0 < a \leq 1$. □

**Proposition 8.6** (Flattening Rank Is a Spectral Point)**.** *Let $V = I_1 \sqcup I_2$ and let $\varphi(t)$ be the rank of the matrix obtained by regarding $t$ as a map between the grouped index spaces. Then $\varphi$ is a spectral point.*

*Proof.* The flattening of $\mathbf{1}$ is the $1 \times 1$ matrix (1). The flattening of a product $st$ is the Kronecker product of the flattenings, and matrix rank is multiplicative under the Kronecker product. If $s = \sum_j s_j$ with $s_j \preceq t_j$, the rank of the flattening is subadditive under sums and nonincreasing under multiplication by matrices, hence $\varphi(s) \leq \sum_j \varphi(s_j) \leq \sum_j \varphi(t_j)$, because the flattening of $(\bigotimes_v A_v)t$ is $(\bigotimes_{v \in I_1} A_v)\,\mathrm{Flat}(t)\,(\bigotimes_{v \in I_2} A_v)^{\mathsf{T}}$. □

**Inherited Result 8.7** (Existence and Duality)**.** *The existence of spectral points below a given upper functional (the source manuscript's Proposition A.5) and the following form of Strassen's duality (its Theorem A.6) are inherited. Let $C \neq 0$ and let $A_1, \ldots, A_r$ be tensors, not all zero. If every spectral point satisfies $\varphi(C) \leq \max_\beta \varphi(A_\beta)$, then for every $\varepsilon > 0$ and all sufficiently large $N$ the power $C^N$ is covered by at most $2^{\varepsilon N}$ tensors, each of which is a product of $N$ factors drawn from $A_1, \ldots, A_r$.*

These are substantive dependencies, not consequences of the functional axioms alone. Proposition A.5 is proved in the manuscript by Zorn's lemma, and Theorem A.6 by applying Proposition A.5 to a function defined through certificates in which factors $C$ are exchanged for factors $A_\beta$. Both remain at inherited status until independently reconstructed.

**Proposition 8.8** (Soundness Versus Completeness of Spectral Obstructions)**.** *Consider the statements (i) $s \preceq t$ implies $\varphi(s) \leq \varphi(t)$ for every spectral point; (ii) $\varphi(s) > \varphi(t)$ for some $\varphi$ implies $s \not\preceq t$; (iii) $s \not\preceq t$ implies the existence of $\varphi$ with $\varphi(s) > \varphi(t)$. Statements (i) and (ii) hold by monotonicity and are equivalent. Statement (iii) is a separation theorem that does not follow from the axioms and is not asserted for ordinary restriction.*

*Proof.* Statement (i) is monotonicity and (ii) is its contrapositive. No derivation of (iii) from the axioms is available, since the axioms are satisfied by families of functionals that do not separate: the single spectral point equal to 1 on every nonzero tensor satisfies them and separates no pair of nonzero tensors. □

A sound obstruction witness is therefore distinguished from a complete characterization of admissibility, and the later theory uses only soundness unless a separation theorem is supplied. The existence and duality theorem of Inherited Result 8.7 is the dependency that guarantees a family of spectral points rich enough for the particular obstructions used by the source, and the elementary axioms alone do not.

# Chapter 9

# Wires, Distances, and the Emergence of Geometry

## 9.1 Elementary Tensor Connections

The simplest tensor that connects two parties is the wire. For parties $u \neq v$ and an integer $q \geq 1$, the wire of dimension $q$ is

$$\langle q \rangle_{uv} = \sum_{i=1}^{q} e_i^{(u)} \otimes e_i^{(v)},$$

the identity matrix of size $q \times q$ in the indices of $u$ and $v$, all other parties having trivial dimension [30]. In the literature on graph tensors a wire is an entangled pair of level $q$, and in tensor networks it is a bond of dimension $q$. Since $\langle q \rangle \langle q' \rangle = \langle qq' \rangle$ on the same pair, a wire of dimension $2^n$ is a product of $n$ wires of dimension two, and it is said to carry $n$ bits.

## 9.2 Spectral Exponents

The value of a spectral point on a wire is determined up to one real number. For every spectral point $\varphi$ and parties $u, v$ there is $d_\varphi(u, v) \geq 0$ with

$$\varphi(\langle q \rangle_{uv}) = q^{d_\varphi(u,v)} \qquad \text{for all } q \geq 1,$$

and the function $d_\varphi$, extended by $d_\varphi(u, u) = 0$, is a pseudometric on the set of parties with values in $[0, 1]$ [30]. The exponent is what $\varphi$ charges for one bit of a wire between the two parties, which explains why the pseudometrics induced in this way are called prices.

## 9.3 Geometry as a Derived Rather Than Imposed Structure

The distance $d_\varphi(u, v)$ is not supplied by a pre-existing physical embedding of the parties. It arises from the transformation properties of the tensors together with the functional, and it satisfies the axioms of a pseudometric because of restriction relations among wires. The triangle inequality, for instance, holds because a wire between $u$ and $w$ is a restriction of the product of a wire between $u$ and $v$ and a wire between $v$ and $w$, obtained by contracting the two indices at $v$. This is a particularly important conceptual feature of the construction: computational relations generate their own geometry, and the geometry carries the information that the functional extracts.

## 9.4 Network Length

For a network

$$t = \prod_{e \in E} \langle q_e \rangle_{u_e v_e},$$

multiplicativity and the wire exponent identity give

$$\log \varphi(t) = \sum_{e \in E} d_\varphi(u_e, v_e) \log q_e.$$

The logarithmic wire dimension $\log q_e$ plays the role of a weight, the spectral pseudometric supplies the geometry, and the value of the functional is a weighted network length. Products of wires are the only tensors whose value can be read directly from the metric, and this is why the argument works with polynomials whose coefficient tensors are products of wires [30].

## 9.5 Distinguishing Spectral Length from Physical Distance

The analogy between spectral network length and communication bandwidth, physical wiring, energy expenditure, or execution time is suggestive and must be handled with restraint. The quantities may share mathematical structure, in that each is a sum over connections of a weight times a distance, but the original theorem establishes no equivalence among them. A spectral point is a functional on tensors, not a measurement of any physical system, and the lengths it assigns are consequences of its axioms. Any application to physical computing architectures requires additional modeling assumptions that connect tensor dimensions to actual communication resources, and these assumptions would constitute separate hypotheses.

## 9.6 Formal Statements

**Definition 9.1** (Wire Tensor)**.** For parties $u \neq v$ and $q \geq 1$, $W_q(u,v) = \sum_{i=1}^{q} e_i^{(u)} \otimes e_i^{(v)}$, with every other party having a one-element index set.

**Proposition 9.2** (Multiplicative Wire Composition)**.** *Up to local changes of basis,*

$$W_p(u,v)\, W_q(u,v) \simeq W_{pq}(u,v).$$

*Consequently $\varphi(W_{pq}) = \varphi(W_p)\varphi(W_q)$ for every spectral point.*

*Proof.* At each of the two parties the index space of the product is $\mathbb{F}^p \otimes \mathbb{F}^q \cong \mathbb{F}^{pq}$ with basis $e_i \otimes e_j$, and the product equals $\sum_{i,j} (e_i \otimes e_j)^{(u)} \otimes (e_i \otimes e_j)^{(v)}$, which is $W_{pq}(u,v)$ after identifying the basis pairs with a basis of $\mathbb{F}^{pq}$. The identification is a local isomorphism at each party, so the two tensors are restrictions of each other. Multiplicativity of $\varphi$ gives the second claim. □

**Inherited Result 9.3** (Spectral Exponents)**.** *For every spectral point $\varphi$ there is a pseudo-metric $d_\varphi$ on the parties with values in $[0,1]$ such that $\varphi(W_q(u,v)) = q^{d_\varphi(u,v)}$ for all $q \geq 1$. That is, $d_\varphi(u,u) = 0$, $d_\varphi(u,v) = d_\varphi(v,u)$, and $d_\varphi(u,w) \leq d_\varphi(u,v) + d_\varphi(v,w)$. This is the manuscript's Lemma A.2.*

The manuscript's argument is short: the map $q \mapsto \varphi(W_q)$ is at least 1, multiplicative, and nondecreasing, because $W_q$ is a restriction of $W_{q'}$ for $q \leq q'$, and a function with these properties is a power of $q$. A wire of dimension $q$ is covered by $q$ copies of $\mathbf{1}$, whence $d_\varphi \leq 1$, and the triangle inequality follows from the restriction of $W_q(u,w)$ from $W_q(u,v)W_q(v,w)$. The argument is written out in full from the axioms of a spectral point in Appendix C (Proposition C.2).

**Definition 9.4** (Spectral Network Length)**.** For a weighted graph $G = (V, E, q)$ define $T_G = \prod_{e \in E} W_{q_e}(u_e, v_e)$ and $\mathcal{L}_\varphi(G) = \log \varphi(T_G)$.

**Proposition 9.5** (Additive Length Identity)**.** $\mathcal{L}_\varphi(G) = \sum_{e \in E} d_\varphi(u_e, v_e) \log q_e$.

*Proof.* Multiplicativity gives $\varphi(T_G) = \prod_e \varphi(W_{q_e}(u_e, v_e))$, and the wire exponent identity gives each factor as $q_e^{d_\varphi(u_e,v_e)}$. Taking logarithms yields the sum. □

**Proposition 9.6** (Cut Rank of Graph Tensors)**.** *Let $T_G = \prod_{e \in E} W_{q_e}(u_e, v_e)$ and let $S \subseteq V$. Then $r_S(T_G) = \prod_{e \in \delta(S)} q_e$, where $\delta(S)$ is the set of edges with exactly one endpoint in $S$. Consequently $\log r_S(T_G) = \sum_{e \in \delta(S)} \log q_e$.*

*Proof.* Order the parties as $S$ followed by $S^c$ and group the edges into those inside $S$, inside $S^c$, and crossing. The tensor is the product of the three groups. The flattening of an edge inside $S$ or inside $S^c$ is a vector (rows indexed by $S$ and a single column, or the reverse), hence rank one. The flattening of a crossing edge is an identity matrix of size $q_e$. The flattening of a product of tensors on disjoint sets of parties is the Kronecker product of their flattenings, and matrix rank is multiplicative under the Kronecker product, which gives $\prod_{e \in \delta(S)} q_e$. □

The identity connects tensor dependencies to weighted graph cuts without any use of the principal separation theorem, and it serves as a recurring worked example: the flattening ranks across the balanced cuts of a graph tensor are the capacities of those cuts.

# Chapter 10

# Rank, Partitions, and Hub Decompositions

## 10.1 Tensor Flattenings

Let $V = I_1 \sqcup I_2$ be a partition of the parties. A tensor $t$ on $V$ can be regarded as a matrix whose rows are indexed by the combined indices of the parties in $I_1$ and whose columns are indexed by those of the parties in $I_2$. The rank of this matrix is the rank of $t$ across the partition, and the matrix is called a flattening. A flattening reorganizes the data of the tensor into a form to which the machinery of matrix rank applies, at the price of forgetting how the indices within each group are related to one another.

## 10.2 Rank as a Limitation on Cross-Partition Dependence

If the rank across the partition is at most $r$, then the tensor admits a decomposition

$$t = \sum_{j=1}^{r} a_j \otimes b_j,$$

where each $a_j$ is a tensor on $I_1$ and each $b_j$ is a tensor on $I_2$. The number $r$ is the number of independent channels through which the two groups of parties are coupled. A rank of one means that the two groups are independent, and a large rank means that the tensor encodes many independent correlations between them. This representation limits the information that must pass between the groups.

## 10.3 The Two-Hub Construction

The decomposition can be realized by a network. Introduce two auxiliary parties $o_1, o_2$ called hubs, connect each party $p \in I_1$ to $o_1$ by a wire of dimension $n_p$, each party $p \in I_2$ to $o_2$ by a wire of the same kind, and join the hubs by a wire of dimension $r$. The original

tensor is a restriction of the resulting product of wires, and consequently, for every spectral point on the enlarged set of parties,

$$\begin{aligned}\log \varphi(t) \leq & \sum_{p \in I_1} d_\varphi(p, o_1) \log n_p \\ & + \sum_{p \in I_2} d_\varphi(p, o_2) \log n_p \\ & + d_\varphi(o_1, o_2) \log r.\end{aligned}$$

This is the central mechanism by which the manuscript translates rank into geometrical length [30]. The rank enters only through the last term.

## 10.4 Bottlenecks and Interface Complexity

The hub connection is an interface through which every cross-partition dependency must pass, and the term $d_\varphi(o_1, o_2) \log r$ is the cost of that interface. The decomposition separates two kinds of complexity: the complexity internal to each component, which is represented by the wires from the parties to their hub, and the complexity required to preserve compatibility between components, which is represented by the hub wire. A tensor that is expensive but admits a decomposition with a narrow interface is expensive only internally, and a tensor whose every decomposition has a wide interface is expensive in the relational sense.

## 10.5 From Individual Cuts to Simultaneous Obstruction

Showing that one partition is expensive is insufficient when a circuit may exploit many alternative decompositions. A circuit with $L$ gates need not decompose its output along any single partition, and the lower bound must control a sufficiently broad family of possible decompositions simultaneously. The relevant family in the manuscript consists of the balanced partitions, and a single spectral point must give a uniform gap across all of them, which motivates the balanced-partition argument of Part VI.

## 10.6 Formal Statements

**Definition 10.1** (Flattening Rank)**.** For $S \subseteq V$ let $r_S(t) = \operatorname{rank}\big(\operatorname{Flat}_{S|S^c}(t)\big)$, the rank of $t$ regarded as a matrix from the indices of $S^c$ to those of $S$.

**Proposition 10.2** (Rank Decomposition)**.** *$r_S(t) \leq r$ if and only if $t = \sum_{i=1}^{r} a_i \otimes b_i$, where $a_i$ is a tensor on $S$ and $b_i$ a tensor on $S^c$.*

*Proof.* A matrix has rank at most $r$ exactly when it is a sum of at most $r$ matrices of rank at most one, and a rank-one matrix is an outer product of a column vector indexed by the indices of $S$ and a row vector indexed by those of $S^c$. □

**Proposition 10.3** (Hub Realization)**.** *Let $V = I_1 \sqcup I_2$ and let $t = \sum_{i=1}^{r} \alpha_i \otimes \beta_i$ with $\alpha_i$ a tensor on $I_1$ and $\beta_i$ a tensor on $I_2$. Let $n_p$ be the size of the index set of $t$ at $p$ and let $o_1, o_2 \notin V$ be further parties. Then $t$ is a restriction of*

$$\prod_{p \in I_1} \langle n_p \rangle_{po_1} \cdot \langle r \rangle_{o_1 o_2} \cdot \prod_{p \in I_2} \langle n_p \rangle_{po_2}.$$

*Proof.* The index space of $o_1$ is $\bigotimes_{p \in I_1} \mathbb{F}^{n_p} \otimes \mathbb{F}^r$ and that of $o_2$ is $\bigotimes_{p \in I_2} \mathbb{F}^{n_p} \otimes \mathbb{F}^r$. The product equals $\sum_{j,j',k} e_j^{(I_1)} \otimes e_j^{(o_1)} \otimes e_k^{(o_1)} \otimes e_k^{(o_2)} \otimes e_{j'}^{(o_2)} \otimes e_{j'}^{(I_2)}$, where $j$ and $j'$ range over multi-indices. Apply the identity at every party of $V$, the linear functional $\lambda_1(e_j \otimes e_k) = \alpha_k[j]$ at $o_1$, and $\lambda_2(e_{j'} \otimes e_k) = \beta_k[j']$ at $o_2$, so that the dimensions at $o_1$ and $o_2$ become one. The result is $\sum_{j,j',k} \alpha_k[j]\beta_k[j'] \, e_j \otimes e_{j'} = \sum_k \alpha_k \otimes \beta_k = t$. □

The source states the lemma in a form in which $o_1$ and $o_2$ may also be given parties of the network, which is used later and is not reproved here. The argument above gives the version with two new hub parties.

**Corollary 10.4** (Rank-to-Distance Inequality)**.** *For a spectral point $\varphi$ on the enlarged party set and a tensor $t$ with $r_S(t) \leq r$,*

$$\begin{aligned} \log \varphi(t) \leq & \sum_{v \in S} d_\varphi(v, o_1) \log n_v \\ & + \sum_{v \in S^c} d_\varphi(v, o_2) \log n_v \\ & + d_\varphi(o_1, o_2) \log r. \end{aligned}$$

*Proof.* By Proposition 10.3 and monotonicity, $\varphi(t)$ is at most the value of $\varphi$ on the product of wires. By multiplicativity and the wire exponent identity this value is the sum on the right, after taking logarithms. □

**Proposition 10.5** (Geometric Rank Obstruction)**.** *Let $t \neq 0$, let all wire dimensions be positive, and let $B_\varphi(S) = \sum_{v \in S} d_\varphi(v, o_1) \log n_v + \sum_{v \in S^c} d_\varphi(v, o_2) \log n_v$. If $d_\varphi(o_1, o_2) > 0$, then*

$$\log r \geq \frac{\log \varphi(t) - B_\varphi(S)}{d_\varphi(o_1, o_2)}.$$

*Proof.* Rearrange the preceding corollary, dividing by the positive number $d_\varphi(o_1, o_2)$. □

This is the first explicit formula converting a geometric discrepancy into a lower bound on a computational resource.

The bound is informative only when $\log \varphi(t) > B_\varphi(S)$: a negative numerator gives a vacuous lower bound, and the hypotheses $t \neq 0$ and $d_\varphi(o_1, o_2) > 0$ are needed so that logarithms and the division are defined.

# Part V

# Curvature and the Geometry of Computational Obstruction

# Chapter 11

# The Crossing-Wire Identity

## 11.1 Crossing Dependencies

Consider four parties $a, b, c, d$ and the product of two wires, one connecting $a$ to $c$ and one connecting $b$ to $d$. Two partitions of the four parties into pairs are in competition: $ab \mid cd$ and $ad \mid bc$. A wire between $a$ and $c$ crosses the first partition and also the second, and likewise for the wire between $b$ and $d$. Flattening the product across either partition gives a permutation matrix of size $q_1 q_2$ when the wires have dimensions $q_1$ and $q_2$, so that the rank across both partitions is full. The hub inequality of the previous chapter therefore says nothing useful about the product, since the rank term it involves is as large as possible for either partition.

## 11.2 Competing Decompositions

The manuscript overcomes the difficulty with an identity between integral tensors, its Proposition B.2, which writes the product of two wires as a sum of a small number of tensors, each of which has small rank across one of the two partitions precisely when it has large rank across the other. The statement involves Frobenius symbols. A symbol is a pair $(I, J)$ of finite subsets of $\mathbb{N}$ with $|I| = |J|$, its size is $|I|$, and its energy is $\sum_{i \in I}(2i+1) + \sum_{j \in J}(2j+1)$. Let $P(E)$ be the number of symbols of energy at most $E$. Fix sizes $n_1, n_2$ and energies $e_1, e_2$, and let $\Sigma_1, \Sigma_2$ be the sets of symbols of size and energy $(n_1, e_1)$ and $(n_2, e_2)$, assumed nonempty. Tensors on four parties have index set $\Sigma_1$ at $a$ and $c$ and $\Sigma_2$ at $b$ and $d$.

The identity asserts the existence, for $\Theta = 2(e_1 + e_2)$, of integral tensors $\Psi_0, \ldots, \Psi_\Theta$ such that their sum is $\langle |\Sigma_1| \rangle_{ac} \langle |\Sigma_2| \rangle_{bd}$, such that $\Psi_N$ is a sum of at most $(N+1)P(N)$ products of a function of $(s_a, s_d)$ and a function of $(s_b, s_c)$, and such that $\Psi_N$ is a sum of at most $(N'+1)P(N')$ products of a function of $(s_a, s_b)$ and a function of $(s_c, s_d)$, where $N' = \Theta - N$ [30]. The two bounds on the number of products trade off, since $P(E)$ grows like $\exp(\kappa\sqrt{E})$.

The tensors $\Psi_N$ are, up to a sign, the coefficients of the determinant $\Psi = \det M$ of a square matrix $M$ with entries in $\mathbb{Z}[X]$, where $y = X^2$. The entries involve the polynomials $B(r, n)$ and $C(r, n)$, the coefficients of $t^n$ in $(1+t)^r/(1+yt)^{r+1}$ and $(1+yt)^r/(1+t)^{r+1}$,

which are Meixner polynomials and satisfy the truncated orthogonality relation

$$(1-y)\sum_{r<R} y^r B(r,i)B(r,j) \equiv [i=j]\, y^i \pmod{y^R}.$$

The rows of $M$ are indexed by the elements of $I_a, I_d, I_c, I_b$ and the columns by those of $J_a, J_d, J_c, J_b$, with the following block entries for a row index $i$ and a column index $j$, read from the printed table of the manuscript.

$$M = \begin{pmatrix} 0 & C(i,j) & [i{=}j]X^{2i+1} & (y{-}1)X^{2i+1}B(i,j) \\ -C(j,i) & 0 & X^{2j+1}B(j,i) & -[i{=}j]X^{2i+1} \\ -[i{=}j]X^{2i+1} & -X^{2i+1}B(i,j) & 0 & -(y{-}1)C(i,j) \\ -(y{-}1)X^{2j+1}B(j,i) & [i{=}j]X^{2i+1} & (y{-}1)C(j,i) & 0 \end{pmatrix}$$

with block rows and columns in the order $a, d, c, b$. At $X = 1$ every entry with a factor $y-1$ vanishes and the determinant collapses to a sign times an indicator, which gives the first property. The degree bound and the second tradeoff come from a reflection identity

$$X^{\Theta}\Psi(1/X; s_a, s_b, s_c, s_d) = \Psi(X; s_a, s_d, s_c, s_b),$$

and the first trade-off from a Schur complement argument that exhibits each coefficient as a sum of products factoring across the pair $\{a,d\} \mid \{b,c\}$.

## 11.3 The Conversion of Algebraic Identities into Geometric Inequalities

Applying the hub inequality to the decomposition for each of the two partitions, and choosing the dominant summand, yields an inequality among the distances between the four original parties and the four hubs. The manuscript states it as its Proposition B.3. For points $x, v, y, u, o, o', w, w'$ of a price and numbers $p, q \geq 0$ there are $r, r' \geq 0$ with $r^2 + r'^2 \leq 2(p^2+q^2)$ such that $p|xv| + q|yu|$ is at most each of

$$p|xo| + q|yo| + p|vo'| + q|uo'| + r|oo'|$$

and

$$p|xw| + q|uw| + q|yw'| + p|vw'| + r'|ww'|,$$

and consequently, for every $t_0 > 0$,

$$\begin{aligned} 2(p|xv| + q|yu|) \leq\ & p(|xo| + |vo'| + |xw| + |vw'|) \\ & + q(|yo| + |uo'| + |uw| + |yw'|) \\ & + t_0(p^2+q^2) + \frac{|oo'|^2 + |ww'|^2}{2t_0}. \end{aligned}$$

Here $x, v, y, u$ play the roles of $a, c, b, d$, and the numbers $r, r'$ measure the dimensions of the two hub wires on a logarithmic scale. A relationship between tensor ranks has become a restriction on the distances among the original parties and their auxiliary hubs.

## 11.4 Why Crossing Structures Matter

The mechanism suggests an interpretation of how computational obstruction arises. A system may be locally simple under several separate descriptions, each of which decomposes it efficiently along one partition, and still resist any single globally efficient description, because the decompositions are incompatible: the dependencies that one decomposition places inside a component are the dependencies that another places across the interface. The crossing identity gives a quantitative form of this incompatibility, since the pieces that are cheap on one side are expensive on the other, and the resulting inequality constrains every metric that assigns a consistent price to both. This is offered as a candidate interpretation and not as a theorem about arbitrary systems.

## 11.5 Limits of the Crossing Analogy

The obstruction concerns algebraic tensor transformations. It does not concern planar graph crossings or the physical crossing of communication wires. The two wires in the identity cross only in the sense that their endpoints interleave in the two competing pairings of four parties, and the geometric representation of Chapter 9 should not be confused with a literal hardware layout.

## 11.6 Formal Statements

**Definition 11.1** (Crossing Tensor)**.** For $q_1, q_2 \geq 1$ let $X_{q_1,q_2} = W_{q_1}(a, c)\, W_{q_2}(b, d)$ on the parties $a, b, c, d$. Write $X_q = X_{q,q}$.

**Proposition 11.2** (Competing Flattening Ranks)**.** $r_{ab|cd}(X_{q_1,q_2}) = r_{ad|bc}(X_{q_1,q_2}) = q_1 q_2$. *In particular* $r_{ab|cd}(X_q) = r_{ad|bc}(X_q) = q^2$. *The third partition has* $r_{ac|bd}(X_{q_1,q_2}) = 1$.

*Proof.* Write $X_{q_1,q_2} = \sum_{i \leq q_1,\, j \leq q_2} e_i^{(a)} \otimes e_i^{(c)} \otimes e_j^{(b)} \otimes e_j^{(d)}$. For $ab \mid cd$ the row index is $(i, j)$ and the column index is $(i, j)$, so the flattening is an identity matrix of size $q_1 q_2$. For $ad \mid bc$ the nonzero entries sit at the row index $(a, d) = (i, j)$ and the column index $(b, c) = (j, i)$, so the flattening is a permutation matrix of the same size. For $ac \mid bd$ the tensor is the product of the vector $\sum_i e_i \otimes e_i$ on $\{a, c\}$ and the vector $\sum_j e_j \otimes e_j$ on $\{b, d\}$, which has rank one. □

**Inherited Result 11.3** (Crossing Identity)**.** *The statement of Proposition B.2 given in Section 11.2 above, with the symbol classes, the bounds* $(N + 1)P(N)$ *and* $(N' + 1)P(N')$*, and the value* $\Theta = 2(e_1 + e_2)$*, is inherited. Its proof by the determinant* $\det M$ *and the Meixner orthogonality congruence must be independently reconstructed.*

**Proposition 11.4** (Dominant Spectral Summand)**.** *If $X = \sum_{j \in J} X_j$ is a finite decomposition, then $\varphi(X) \le |J| \max_j \varphi(X_j)$. Therefore some $j_*$ satisfies $\log \varphi(X_{j_*}) \ge \log \varphi(X) - \log |J|$.*

*Proof.* Each $X_j$ is a restriction of itself, so the cover axiom gives $\varphi(X) \le \sum_j \varphi(X_j) \le |J| \max_j \varphi(X_j)$. Taking the logarithm of the maximum gives the second claim. □

Applying the hub inequality to the dominant summand under both competing partitions produces the required crossing constraints.

**Inherited Result 11.5** (Crossing Inequality)**.** *The inequality displayed in the previous section, the manuscript's Proposition B.3 with its consequence (B.2), is inherited.*

**Computational check of the identity.** The determinant matrix $M$ was constructed exactly over the integers for four small parameter classes, namely $(n_1, e_1, n_2, e_2) = (1, 2, 1, 2)$, $(1, 4, 1, 2)$, $(1, 4, 1, 4)$ and $(2, 6, 1, 2)$, all quadruples of symbols were enumerated, and $\det M$ was computed symbolically in $X$. In every case the degree of $\Psi$ did not exceed $\Theta$, the reflection identity held exactly, the value at $X = 1$ was supported precisely on the quadruples satisfying the matching conditions $J_c = I_a$, $I_c = J_a$, $J_b = I_d$, $I_b = J_d$ with the constant value 1, and the rank of every coefficient flattening across the two competing partitions respected the bound $(N + 1)P(N)$. This is an exact finite check on small cases and is not a proof of the identity. The discriminating power of the individual checks was tested by deliberately corrupting the matrix: flipping the sign of one block, replacing the factor $y - 1$ by $y + 1$ in one block, and transposing the arguments of $C$ in one block. In the three smallest parameter classes the reflection identity or the value of the determinant at $X = 1$ detected every corruption except one, the transposed arguments of $C$, which passed both checks in the smallest class and was detected in the other two. The coefficient rank bound was never violated, by the original or by any corrupted matrix, so it did not discriminate in these small cases, most likely because the bound $(N + 1)P(N)$ is large relative to the dimensions of the flattenings, although this was not measured. The evidence rests on the degree, reflection, and specialization checks. A meaningful test of the rank bound would require cases in which the ratio of the flattening rank of $\Psi_N$ to $(N + 1)P(N)$ can approach one, and none is available at the sizes computed. The exact enumeration scripts are not reproduced in this version.

# Chapter 12

# Nonpositive Curvature

## 12.1 Metric Midpoints

A point $y$ is a midpoint of $x$ and $x'$ if $d(x,y) = d(y,x') = \frac{1}{2}d(x,x')$. In a Euclidean space every pair has exactly one midpoint. In a general metric space midpoints may fail to exist, may be non-unique, or may exist for a dense set of pairs only. The conditions required to extend a finite pseudometric structure to a complete metric space in which midpoints exist are part of the machinery of the manuscript's Appendix C, which constructs the required enlargement by adjoining random points whose distances are bounded by expected distances.

## 12.2 The Bruhat–Tits Inequality

A metric space satisfies the inequality of Bruhat and Tits, also called the CN inequality, if for all points $x, x', z$ and every midpoint $y$ of $x$ and $x'$,

$$d(z,y)^2 \le \tfrac{1}{2}d(z,x)^2 + \tfrac{1}{2}d(z,x')^2 - \tfrac{1}{4}d(x,x')^2.$$

The inequality is a nonpositive curvature condition. It says that a median of a triangle is no longer than in the Euclidean triangle with the same side lengths, which is the form in which negative or zero curvature is expressed for geodesic metric spaces without reference to differentiable structure [9, 8].

## 12.3 The Euclidean Equality Case

In Euclidean space the inequality holds with equality. For $y = \frac{1}{2}(x + x')$ one expands

$$\left\| z - \tfrac{x+x'}{2} \right\|^2 = \|z\|^2 - \langle z, x + x' \rangle + \tfrac{1}{4}\|x + x'\|^2,$$

and the right side of the claimed identity expands to the same expression, using $\|x + x'\|^2 = 2\|x\|^2 + 2\|x'\|^2 - \|x - x'\|^2$. The elementary derivation gives an accessible bridge between

linear algebra and the abstract tensor geometry: the curvature inequality is the parallelogram law of the inner product, relaxed to an inequality.

## 12.4 CAT(0) Spaces

A complete metric space in which any two points have a midpoint satisfying the inequality above is a CAT(0) space [8]. Such spaces are geodesic, uniquely so, and distances to a convex set behave convexly. The manuscript uses the crossing-wire argument to show that every price satisfies the Bruhat–Tits inequality, and then that every price is the pull-back of the metric of a complete CAT(0) space whose metric is again a price [30]. The identification matters because the standard toolkit for CAT(0) geometry then becomes available for the analysis of prices.

## 12.5 Curvature as a Constraint on Computational Organization

The inequality constrains which families of distances can arise from spectral points, and therefore which decompositions of tensors are simultaneously possible. The question for the remainder of the argument is whether the geometry of admissible transformations forces certain networks to remain expensive under every permitted decomposition. In nonpositive curvature the answer depends on how nearly flat the available metrics are, and the dichotomy of Chapter 13 records the two possibilities.

## 12.6 Formal Statements

**Definition 12.1** (Metric Midpoint)**.** A point $y$ is a midpoint of $x$ and $x'$ if $d(x,y) = d(y,x') = \frac{1}{2}d(x,x')$.

**Proposition 12.2** (Euclidean Midpoint Identity)**.** *If $y = \frac{1}{2}(x+x')$ in a Euclidean space, then*

$$\|z-y\|^2 = \tfrac{1}{2}\|z-x\|^2 + \tfrac{1}{2}\|z-x'\|^2 - \tfrac{1}{4}\|x-x'\|^2.$$

*Proof.* Write $z - y = \frac{1}{2}(z-x) + \frac{1}{2}(z-x')$ and expand the squared norm with the inner product: $\|z-y\|^2 = \frac{1}{4}\|z-x\|^2 + \frac{1}{4}\|z-x'\|^2 + \frac{1}{2}\langle z-x, z-x'\rangle$. Since $2\langle z-x, z-x'\rangle = \|z-x\|^2 + \|z-x'\|^2 - \|x-x'\|^2$, the right side equals $\frac{1}{2}\|z-x\|^2 + \frac{1}{2}\|z-x'\|^2 - \frac{1}{4}\|x-x'\|^2$. □

**Inherited Result 12.3** (Curvature Inequality for Prices)**.** *Let $x, v, m, z$ be points of a price with $|xm|, |vm| \le D$ and $|xv| = 2D$. Then $|zm|^2 \le \frac{1}{2}|zx|^2 + \frac{1}{2}|zv|^2 - D^2$. This is Theorem C.1 of the source manuscript, and it is the Bruhat–Tits inequality for midpoints of prices.*

**Proposition 12.4** (Uniqueness of Midpoints)**.** *In a pseudometric space satisfying the curvature inequality, if $y$ and $y'$ are both midpoints of $x$ and $x'$, then $d(y, y') = 0$. In a metric space, $y = y'$.*

*Proof.* Apply the inequality with $z = y'$ and midpoint $y$. Since $d(y', x) = d(y', x') = \frac{1}{2}d(x, x')$,

$$d(y', y)^2 \leq \tfrac{1}{2} \cdot \tfrac{d(x,x')^2}{4} + \tfrac{1}{2} \cdot \tfrac{d(x,x')^2}{4} - \tfrac{1}{4}d(x, x')^2 = 0.$$

□

**Inherited Result 12.5** (CAT(0) Realization)**.** *Every price is the pull-back of the metric of a complete CAT(0) space whose metric is again a price (the manuscript's Proposition C.3, with the extension and completion constructions of Appendix C and the geometric toolkit of Lemma C.4).*

*Remark* 12.6. Two cautions apply to the passage from prices to CAT(0) spaces. First, the reconstruction must distinguish quotienting a pseudometric by zero distance, adjoining points, constructing exact or approximate midpoints, and taking metric completions, and it must show that each operation keeps the metric inside the class of prices; an embedding into a CAT(0) space is not sufficient if the enlarged metric no longer represents a spectral price. Second, nonpositive curvature is not an automatic consequence of distributed dependence: a general resource preorder can induce distances that are not CAT(0), and the curvature result rests on specific algebraic identities and closure properties.

# Chapter 13

# The Flat and Contracting Alternatives

## 13.1 Spectral Prices

A price on a finite set $V$ is a pointwise limit of functions $c\,d_{\varphi_j}$, where $c \geq 0$ is a real number and $\varphi_j$ are spectral points. A function on the pairs of an arbitrary set is a price if all of its pull-backs to finite sets are prices [30]. Allowing scaling and limits enlarges the available geometries: nonnegative multiples, pull-backs, and pointwise limits of prices are again prices, and the prices bounded by a given function form a compact set by Tychonoff's theorem. Compactness is what allows the argument to pass from a family of approximate configurations to an exact one.

## 13.2 The Contracting Hypothesis

The first of the two technical hypotheses involves the set $V = \mathrm{Mat}_{\mu}(\mathbb{F}_2)$ of $\mu \times \mu$ matrices over the field with two elements. This set only supplies the parties and their addition, and its field $\mathbb{F}_2$ is not the field over which the tensors are defined. For a price $d$ on $V$ let

$$D(r)^2 = \mathbb{E}_{\substack{a,W \\ \mathrm{rank}(W)=r}} d(a, a+W)^2,$$

the mean over all matrices $a$ and all matrices $W$ of rank $r$. The contracting hypothesis asserts the existence of an integer $\mu \geq 2$ and a real number $s$ with $1 \leq s$ and $s^2 < \mu$ such that $D(\mu) \leq s\,D(1)$ for every price on $\mathrm{Mat}_{\mu}(\mathbb{F}_2)$. Every price satisfies $D(\mu) \leq \sqrt{\mu}\,D(1)$, by an argument with quadrilaterals whose squared diagonals add up to at most the sum of the squared sides. The hypothesis therefore says that, for some $\mu$, every price falls short of a Euclidean rectangle by a fixed factor.

## 13.3 The Flat Hypothesis

The alternative hypothesis states that Euclidean geometry is available: all finite subsets of all Euclidean spaces, with their metrics, are prices. Equivalently, for every finite subset $V$ of a Euclidean space and every $\varepsilon > 0$ there are a spectral point $\varphi$ on $V$ and a real number $c \geq 0$ with

$$\left| c\, d_\varphi(u, v) - \|u - v\| \right| \leq \varepsilon$$

for all $u, v \in V$. In the flat case prices can be Euclidean, and this is what permits the explicit configurations of Part VI.

## 13.4 Exhaustiveness Without Exclusivity

The manuscript proves that at least one of the two hypotheses holds over every field, and states plainly that it does not determine which holds even over the real numbers and does not show that the two are mutually exclusive [30]. The proof assumes that the contracting hypothesis fails and derives the flat one. A compactness argument produces prices on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ with $D(\mu)^2 \geq \mu$ and $D(1) = 1$, and subadditivity then forces equality, from which it follows that squared distances in this price equal ranks of differences, so that the diagonal matrices with entries in $\{0, 1\}$ form a Euclidean cube, which is then filled using the geometry of CAT(0) spaces. This is a particular form of mathematical argument: a conclusion is established through alternative regimes without identifying the regime that is actually realized.

## 13.5 The Epistemological Significance of Unresolved Alternatives

Let $A$ and $B$ be two hypotheses and $C$ a conclusion. The proposition

$$(A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)$$

establishes $C$ by cases and is logically weaker, in the information it carries about the underlying mathematical universe, than the determination of which of $A$ or $B$ holds. The former is a complete logical result. The latter would be an explanatory result, since knowing the regime would indicate what the relevant geometry actually is. The theorem here is complete as a logical result while incomplete as an explanatory or constructive theory, and the distinction recurs in Part IX, where the unknown value of $\mu$ is one reason that no explicit threshold is available.

## 13.6 Formal Statements

**Definition 13.1** (Price Space)**.** Let $\mathcal{D}_V$ be the closure, in the space of pairwise distance functions on $V$ with pointwise convergence, of $\{c\, d_\varphi : c \geq 0,\ \varphi \text{ a spectral point on } V\}$.

**Definition 13.2** (Mean Rank Displacement)**.** For $V = \mathrm{Mat}_\mu(\mathbb{F}_2)$ and $d \in \mathcal{D}_V$ let $D_d(r)^2 = \mathbb{E}_{a,\, W:\mathrm{rank}\, W = r}\, d(a, a+W)^2$.

**Definition 13.3** (Contracting Condition)**.** $\mathsf{C}(\mathbb{F})$ holds if there are an integer $\mu \geq 2$ and a real $s$ with $1 \leq s$ and $s^2 < \mu$ such that $D_d(\mu)^2 \leq s^2 D_d(1)^2$ for all $d \in \mathcal{D}_{\mathrm{Mat}_\mu(\mathbb{F}_2)}$.

**Definition 13.4** (Flat Condition)**.** $\mathsf{F}(\mathbb{F})$ holds if for every finite subset $V$ of a Euclidean space, the restricted Euclidean metric belongs to $\mathcal{D}_V$.

**Inherited Result 13.5** (Subadditivity of Rank Displacement)**.** *For prices on* $\mathrm{Mat}_\mu(\mathbb{F}_2)$ *and* $i + j \leq \mu$, $D(i+j)^2 \leq D(i)^2 + D(j)^2$. *Consequently* $D(\mu)^2 \leq \mu\, D(1)^2$. *This is Step 1 of the proof of the manuscript's Theorem D.2, and it depends on the quadrilateral inequality of Lemma C.4.*

**Inherited Result 13.6** (Fundamental Alternative)**.** *For every field* $\mathbb{F}$, $\mathsf{F}(\mathbb{F}) \vee \mathsf{C}(\mathbb{F})$. *This is the manuscript's Theorem D.2.*

**Proposition 13.7** (Exhaustive Conditional Reasoning)**.** *If* $\mathsf{F} \vee \mathsf{C}$, $\mathsf{F} \Rightarrow \mathsf{L}$, *and* $\mathsf{C} \Rightarrow \mathsf{L}$, *then* $\mathsf{L}$. *Neither* $\mathsf{F} \wedge \neg\mathsf{C}$ *nor* $\mathsf{C} \wedge \neg\mathsf{F}$ *is required.*

*Proof.* Proof by cases: in each case $\mathsf{L}$ follows, and the disjunction guarantees that one case occurs. No premise about which case occurs, or whether both occur, is used. □

The manuscript leaves the actual regime over the real numbers unresolved. The logical completeness of the lower-bound argument therefore does not require a classification of the underlying geometry.

*Remark* 13.8. The proof that failure of contraction yields the flat condition is compressed in the source, and a reconstruction should separate the normalization of prices, the compactness argument, the saturation of the inequalities, the construction of Euclidean cubes, and the extension to arbitrary finite-dimensional configurations. Closure of prices under scaling and pointwise limits does not give closure under arithmetic averages, so any use of group averaging must be justified by a specific construction. The field over which tensors are defined and the auxiliary field $\mathbb{F}_2$ indexing the matrix configurations remain distinct throughout.

# Part VI

# The Flat-Case Lower Bound

# Chapter 14

# Complete Tensor Networks

## 14.1 The Complete Graph Construction

The lower bound for the flat case is proved for a family of polynomials chosen so that the dependency structure is as symmetric and as dense as a network can be. Fix integers $k \geq 2$ and $\ell \geq 1$ and consider the complete graph on the nodes $1, \ldots, k$. Every edge $ab$ carries a string $\sigma(ab) \in \{0,1\}^{\ell}$ of $\ell$ bits, so that an assignment of strings to all edges is a point of $\{0,1\}^{\ell N}$ with $N = \binom{k}{2}$. The node $a$ sees the strings on the $k-1$ edges that meet it, and the collection of these strings is its local view $\sigma_a = (\sigma(ab))_{b \neq a}$. Nothing in the construction refers to a physical network. The graph is a bookkeeping device that records which local views must agree, and every pair of nodes is required to agree on exactly one string.

The same structure is described in the language of tensors by giving every end of every edge a party of its own. The end $(a, b)$ is the half of the edge $ab$ that lies at the node $a$, so that there are $k(k-1)$ ends, and the end $(a, b)$ carries an index space of dimension $2^{\ell}$. The two ends of an edge are the parties joined by a wire of dimension $2^{\ell}$ in the sense of Chapter 9. Splitting a node into its ends is what makes it possible to speak of the distance between the two halves of one edge, and the geometry of Chapter 16 places every end at a point of its own.

## 14.2 The Network Polynomial

The network polynomial has one variable $x_{a,\rho}$ for each node $a$ and each local view $\rho \in \{0,1\}^{(k-1)\ell}$, so that the variables of the node $a$ form a block of $2^{(k-1)\ell}$ variables, and it sums over all assignments of strings to edges the product of the variables selected by the local views,

$$f_{k,\ell} = \sum_{\sigma} \prod_{a=1}^{k} x_{a,\sigma_a}.$$

The polynomial is linear in each block, which is the property called set-multilinearity, and its degree is the number of nodes. For $k = 3$ and $\ell = 1$ it is the trilinear form

$\sum_{p,q,r \in \{0,1\}} x_{1,pq} x_{2,pr} x_{3,qr}$, which is up to the names of the variables the trilinear form of the multiplication of two by two matrices, a coincidence that indicates how the construction generalizes a familiar object [30]. The role of the polynomial in the argument is that it is simultaneously easy to describe, since it is a sum over bit strings of products of one variable per node, and difficult to compute, since the sum ranges over exponentially many assignments whose products overlap in a way that no single decomposition of the nodes accommodates.

The polynomial is also expressible as a permanent of moderate size. A sum over bit strings of products of local weights is a clause sum in the terminology of the source, and clause sums are permanents of matrices whose size is a small multiple of the number of literals, with entries that are affine in one variable each. For the network polynomial the number of literals is $k(k-1)\ell\, 2^{(k-1)\ell}$, and the consequence, taken as an inherited result, is that a circuit of size $L$ for the permanent of a sufficiently large matrix yields a circuit of size $5L$ for $f_{k,\ell}$ [30]. The reduction runs in the direction needed for a lower bound: a lower bound for $f_{k,\ell}$ becomes a lower bound for the permanent.

## 14.3 The Coefficient Tensor

The coefficients of a set-multilinear polynomial form a tensor with one index for each block. For $f_{k,\ell}$ the entry at the local views $(\rho_1, \ldots, \rho_k)$ is one when these views are consistent with a single assignment of strings to the edges and zero otherwise, because a monomial $\prod_a x_{a,\rho_a}$ occurs in the sum exactly when the string that $\rho_a$ places on the edge $ab$ equals the string that $\rho_b$ places on it, for every pair. A product of equalities is a product of wires. Regarded as a tensor on the ends, the coefficient tensor is therefore

$$T_{k,\ell} = \prod_{1 \le a < b \le k} \langle 2^{\ell} \rangle_{(a,b)(b,a)},$$

one wire of dimension $2^{\ell}$ for every edge. This is the point at which the abstract machinery of Part IV and Part V meets a concrete polynomial: the spectral value of the polynomial's coefficient tensor is given by the additive length identity of Chapter 9, and its flattening ranks are given by the cut-rank identity of the same chapter.

The flattening rank across a partition of the nodes into a set $S$ and its complement is the first quantity of independent interest. Every edge that joins $S$ to its complement contributes a factor $2^{\ell}$, and every other edge contributes nothing, so that the rank is $2^{\ell |S| (k - |S|)}$. A balanced partition has $|S|(k-|S|)$ at least of order $k^2$, and the rank is then exponential in $\ell k^2$. This is a statement about the dependencies that cross a partition. It does not by itself bound the size of any circuit, because an arithmetic circuit does not decompose its output along one prescribed partition, and the passage from the flattening ranks to a lower bound is the work of the next two chapters.

## 14.4 Local Assignments and Global Compatibility

The structure of the polynomial is that of a constraint satisfaction problem in which the constraints are equalities between local views. Each node holds a view and each edge demands that its two halves agree. The sum over assignments counts the globally compatible families of views, and the variables weight them. The difficulty of a computation of the polynomial is therefore a difficulty of reconstruction in the sense of Chapter 1: contributions that are individually cheap to describe must be combined across a structure in which every node is constrained by every other. In a path, in which each node meets at most two edges, the combination can be organized as a sequence of contractions and costs little, as the formal statements below record. In the complete graph every balanced division of the nodes cuts a number of edges proportional to $k^2$, and no organization of the contractions avoids carrying all of them across some cut. This is the precise sense in which global compatibility is the source of the obstruction, and it is a property of the dependency structure and not of the number of monomials, since the number of monomials is the same, namely $2^{\ell N}$, whether the edges are arranged into a path-like or a complete pattern of the same total weight.

## 14.5 Formal Statements

**Definition 14.1** (Complete Network)**.** For $k \geq 2$ and $\ell \geq 1$ let $N = \binom{k}{2}$. The parties are the ends $(a, b)$ with $a \neq b$, each with index space $\{0,1\}^\ell$, and the complete network tensor is $T_{k,\ell} = \prod_{a<b} W_{2^\ell}((a,b),(b,a))$. The network polynomial is $f_{k,\ell} = \sum_\sigma \prod_{a=1}^{k} x_{a,\sigma_a}$, the sum over all assignments $\sigma$ of strings in $\{0,1\}^\ell$ to the edges of the complete graph on $[k]$, with $\sigma_a = (\sigma(ab))_{b \neq a}$.

**Proposition 14.2** (Structure of the Network Polynomial)**.** *The polynomial $f_{k,\ell}$ has $k\,2^{(k-1)\ell}$ variables, is linear in each of its $k$ blocks and has degree $k$. It has exactly $2^{\ell N}$ distinct monomials, each with coefficient 1, so that it is nonzero over every field. Its coefficient tensor, regarded as a tensor on the ends, is $T_{k,\ell}$.*

*Proof.* The count of variables and the degree are immediate from the definition. A monomial $\prod_a x_{a,\rho_a}$ arising from $\sigma$ determines $\rho_a = \sigma_a$ for every $a$, and $\sigma_a$ contains $\sigma(ab)$ for every $b \neq a$, so that the monomial determines $\sigma$. Distinct assignments therefore give distinct monomials and no coefficients combine, and each monomial has coefficient 1. There are $2^{\ell N}$ assignments. The entry of the coefficient tensor at $(\rho_1, \ldots, \rho_k)$ is 1 exactly when the string that $\rho_a$ puts on $ab$ equals the string that $\rho_b$ puts on $ab$ for all $a < b$, which is the entry of the product of the wires of dimension $2^\ell$ joining $(a,b)$ and $(b,a)$. □

**Proposition 14.3** (Cut Rank of the Complete Network)**.** *For $S \subseteq [k]$ let $\tilde{S}$ be the set of ends $(a,b)$ with $a \in S$. Then $r_{\tilde{S}}(T_{k,\ell}) = 2^{\ell|S|(k-|S|)}$. If $k/3 \leq |S| \leq 2k/3$, then $\log r_{\tilde{S}}(T_{k,\ell}) \geq \frac{2}{9}\ell k^2$.*

*Proof.* By Proposition 9.6 the logarithm of the rank is $\ell$ times the number of edges $ab$ with exactly one endpoint in $S$, which is $|S|(k-|S|)$. The function $s \mapsto s(k-s)$ is concave

and symmetric about $k/2$, so on $[k/3, 2k/3]$ it is at least its value $\frac{k}{3} \cdot \frac{2k}{3} = \frac{2}{9}k^2$ at the endpoints. □

**Proposition 14.4** (Spectral Value of the Complete Network)**.** *For every spectral point $\varphi$,* $\log \varphi(T_{k,\ell}) = \ell \sum_{a<b} d_\varphi\big((a,b),(b,a)\big)$.

*Proof.* This is the additive length identity for the graph consisting of $N$ disjoint edges, each of weight $2^\ell$, so that $\log q_e = \ell$. □

**Definition 14.5** (Path Network)**.** For $k \geq 2$ and $\ell \geq 1$ let

$$g_{k,\ell} = \sum_{s_1,\dots,s_{k-1} \in \{0,1\}^\ell} x_{1,s_1}\, x_{2,(s_1,s_2)} \cdots x_{k-1,(s_{k-2},s_{k-1})}\, x_{k,s_{k-1}},$$

with one variable for each node and each pair of strings on its incident edges.

**Proposition 14.6** (The Path Is Cheap)**.** $L_{\mathbb{F}}(g_{k,\ell}) \leq 3k\, 2^{2\ell}$ *over every field, and every flattening rank of the coefficient tensor across a set of nodes is at most* $2^{\ell(k-1)}$.

*Proof.* Define $v_1[s_1] = x_{1,s_1}$ and, for $2 \leq a \leq k-1$, $v_a[s_a] = \sum_{s_{a-1}} v_{a-1}[s_{a-1}]\, x_{a,(s_{a-1},s_a)}$, and finally $g_{k,\ell} = \sum_{s_{k-1}} v_{k-1}[s_{k-1}]\, x_{k,s_{k-1}}$. The circuit has at most $k2^{2\ell}$ variable gates, at most $k2^{2\ell}$ product gates, and at most $k2^{2\ell}$ sum gates. The coefficient tensor is a product of wires along the $k-1$ edges of the path, so by Proposition 9.6 its cut rank is $2^{\ell c}$ with $c \leq k-1$ the number of path edges crossing the cut. □

The proposition shows that a polynomial of the same general form, a sum over bit strings of products of local weights, has circuits whose size is polynomial in the number of variables when the dependency graph is a path, and that the corresponding flattening ranks are at most $2^{\ell(k-1)}$ across every partition, in contrast with the complete network, whose balanced flattening ranks are at least $2^{2\ell k^2/9}$.

**Inherited Result 14.7** (Network Polynomial as a Permanent)**.** *Let $2 \neq 0$ in $\mathbb{F}$, $k \geq 2$, $\ell \geq 1$ and $M \geq 3k^3 \ell\, 2^{k\ell}$. If $\mathrm{PERM}_M$ has a circuit over $\mathbb{F}$ with $L$ gates, then $f_{k,\ell}$ has one with $5L$ gates. In every characteristic the same holds for $\mathrm{HC}_M$ with 40 in place of 3 in the condition on $M$. This is Lemma E.3 of the manuscript, resting on Proposition E.1 and Corollary E.2 for the permanent and on Proposition M.2 for the Hamiltonian cycle polynomial.*

The proof in the source writes $f_{k,\ell}$ as a clause sum with $k(k-1)\ell\, 2^{(k-1)\ell} \leq M/3$ literals and applies the construction of the permanent of a matrix whose entries are affine in one variable each, substituting these entries for the variables of the given circuit at a cost of five gates each. The construction in Proposition E.1 is elementary but intricate, and it remains at inherited status.

# Chapter 15

# Balanced Decompositions

## 15.1 Balanced Partitions

A circuit does not choose a partition of the nodes. It computes the polynomial by a sequence of additions and multiplications, and the structure of the computation is whatever the circuit happens to be. The lower bound therefore needs a statement that holds for every circuit, and the statement used is a decomposition of the output into products, indexed by the sets of nodes that a product can separate. Call a set $S \subseteq [k]$ balanced if $k \leq 3|S| \leq 2k$, so that it contains between a third and two thirds of the nodes, and write $\mathcal{B}_k$ for the family of balanced sets. The family has at most $2^k$ members, and this crude bound will suffice, because the number of balanced sets enters the final inequality only through its logarithm, which is at most $k$ bits.

The reason for restricting attention to balanced sets is the same as in the classical depth reduction for arithmetic circuits [19, 36]. In a product gate the two factors divide between them the blocks that the product involves, and one of the factors always involves at least half of them. If this factor involves more than two thirds of the blocks, one descends into it, and the descent continues until the number of blocks lies in the balanced range. The balanced range is the one in which every product is guaranteed to separate a substantial part of the structure from another substantial part, and this is why all the information needed for the lower bound is carried by the balanced cuts, whose capacity is of order $k^2$ in the complete network.

## 15.2 Circuit-Induced Tensor Decomposition

The decomposition lemma states that the part of the output linear in every block is a sum over balanced sets $S$ of at most $L$ products of a polynomial in the blocks of $S$, linear in each, and a polynomial in the remaining blocks, linear in each. Two points deserve emphasis. First, the circuit is arbitrary: its intermediate gates compute polynomials of any shape, including polynomials that are not set-multilinear, and may cancel high-degree terms that are produced and discarded. The lemma handles this by tracking for each gate only the

homogeneous parts of prescribed block degree and by observing that the part of degree one in every block of a set arises from parts of lower block degree of earlier gates, with all other parts irrelevant. Second, the lemma is a statement about polynomials, and the passage to tensors is a restatement: the coefficient tensor of a product of a polynomial in the blocks of $S$ and a polynomial in the other blocks is the tensor product of their coefficient tensors, and splitting the index of each node into the strings on its edges is a local isomorphism. Applied to $f_{k,\ell}$ the lemma therefore gives

$$T_{k,\ell} = \sum_{S \in \mathcal{B}_k} t_S,$$

where $t_S$ is a sum of at most $L$ products of a tensor on the ends at the nodes of $S$ and a tensor on the other ends. This is exactly the form to which the hub realization of Chapter 10 applies, with $L$ as the rank.

## 15.3 The Simultaneous Spectral Inequality

Introduce for every balanced set $S$ two hubs $o_S$ and $o'_S$, regarded as parties distinct from the ends and from the hubs of other balanced sets. Every end $(a, b)$ with $a \in S$ is joined to $o_S$ by a wire of dimension $2^\ell$, every other end is joined to $o'_S$ by a wire of the same dimension, and the hubs are joined by a wire of dimension $L$. Let $T_S$ be the product of these wires. By hub realization, $t_S$ is a restriction of $T_S$, so that every spectral point $\varphi$ satisfies $\varphi(t_S) \le \varphi(T_S)$. The tensor $T_{k,\ell}$ is covered by the at most $2^k$ tensors $t_S$, and therefore

$$\varphi(T_{k,\ell}) \le 2^k \max_S \varphi(T_S),$$

by the cover axiom. Taking logarithms and applying the wire exponent identity to both sides gives an inequality that has the same shape for every spectral point. On the left stands the sum over edges of the distance between the two ends, multiplied by $\ell$. On the right stand the $k$ bits that name a balanced set, then the sum over all ends of the distance from the end to its hub, multiplied by $\ell$, and finally the distance between the hubs multiplied by $\log L$. The inequality holds simultaneously for the hubs of the one balanced set at which the maximum is attained, and since the geometry must work for every circuit and the unknown circuit determines the set, the placement of hubs must be arranged so that every balanced set has a favorable configuration. This is the mathematical hinge of the flat-case argument: a hypothetical small circuit becomes a geometrical inequality that must hold across the balanced decompositions. It is stated with the scale factor $c$ that converts spectral exponents into Euclidean lengths in the formal statements, since the unscaled exponent function is not itself Euclidean.

## 15.4 The Difference Between Complete Networks and Paths

The hinge inequality distinguishes the complete network from the path in a way that no count of monomials could. For a path the balanced sets cut at most $k - 1$ edges in total, the

flattening ranks across every partition are at most $2^{\ell(k-1)}$, and the polynomial has circuits of size $3k2^{2\ell}$ by Proposition 14.6. The ends can be placed so that each edge has a small length and the hubs sit close to the ends they serve, with a cost per balanced set that is small compared to the length of the network. For the complete graph the length of the network is $\ell N$ in the placement of Chapter 16, in which every edge has length one, and the cost of attaching the ends of every balanced set to hubs is smaller than the length of the network by an amount proportional to $\ell k$, so that the difference must be paid by the hub wire of dimension $L$. The same set of formulas therefore gives no lower bound for the path and a large lower bound for the complete network, and what separates them is the number of edges that every balanced cut must sever, which is of order $k$ for the path and of order $k^2$ for the complete graph.

## 15.5 Why the Number of Edges Is Not Sufficient

It would be natural to conjecture that a network with many edges requires large circuits simply because the polynomial has many monomials, and the argument shows that this is the wrong description. The monomial count of $f_{k,\ell}$ is $2^{\ell N}$, which is also the monomial count of any set-multilinear network polynomial with $N$ edges of weight $\ell$ bits each, whatever the arrangement of the edges, and Chapter 5 has already shown that exponentially many monomials are compatible with small circuits. What the argument uses is the capacity of balanced cuts, that is, the minimum over balanced sets of the number of edges severed, together with a geometry in which the ends and hubs can be placed so that the hub attachments of all balanced sets save a fixed fraction of the length of the network. The quantity that matters is therefore a bisection-width-like invariant of the dependency graph, joined to a geometrical inequality, and the analogy with other settings in which a cut capacity limits a computation should be understood as a structural analogy and not as an identification of the present theorem with any classical cut lower bound.

## 15.6 Formal Statements

**Definition 15.1** (Balanced Sets)**.** A set $S \subseteq [k]$ is balanced if $k \le 3|S| \le 2k$. The family of balanced sets is $\mathcal{B}_k$, and $|\mathcal{B}_k| \le 2^k$.

**Proposition 15.2** (Balanced Products)**.** *Let a polynomial over a commutative ring with $k \ge 2$ blocks of variables have a circuit of $L$ gates, and let $f$ be its part of degree one in every block. Then*

$$f = \sum_{S \ balanced} \sum_{j=1}^{L} g_{S,j}\, h_{S,j},$$

*where $g_{S,j}$ is linear in each block of $S$ and contains no other variables, and $h_{S,j}$ is linear in each block outside $S$ and contains no other variables.*

*Proof.* For a gate $\gamma$ and a set $S$ of blocks let $\gamma_S$ be the part of $\gamma$ of degree one in each block of $S$ and degree zero in the others. Because the blocks consist of disjoint sets of variables, the block degrees of a product of polynomials add, and therefore $(\gamma_1\gamma_2)_S = \sum_{A \sqcup B = S} (\gamma_1)_A (\gamma_2)_B$. We show by induction along the circuit the following claim: if $3|S| > 2k$, then $\gamma_S = \sum_{\delta, A} \delta_A \, h_{\delta,A}$, where the sum is over the gates $\delta$ of the circuit and the balanced sets $A \subseteq S$, and $h_{\delta,A}$, which depends on $\gamma$ and $S$, is linear in each block of $S \setminus A$ and contains no variables of other blocks.

A variable or a constant has $\gamma_S = 0$ when $|S| \geq 2$, and $3|S| > 2k \geq 4$ gives $|S| \geq 2$. For a sum $\gamma = \gamma_1 + \gamma_2$ we have $\gamma_S = (\gamma_1)_S + (\gamma_2)_S$ and add the representations. For a product $\gamma = \gamma_1\gamma_2$ consider a term $(\gamma_1)_A (\gamma_2)_B$ with $A \sqcup B = S$ and, after exchanging the roles of the two factors if necessary, $|A| \geq |B|$. Then $|A| \geq |S|/2 > k/3$. If $3|A| \leq 2k$, then $A$ is balanced and the term has the required form with $\delta = \gamma_1$ and $h = (\gamma_2)_B$, which is linear in each block of $B = S \setminus A$. If $3|A| > 2k$, the induction hypothesis applies to the earlier gate $\gamma_1$ and the set $A$, giving $(\gamma_1)_A = \sum \delta_{A'} h_{\delta,A'}$ with $A' \subseteq A$ balanced, and multiplying by $(\gamma_2)_B$ gives the required form with the polynomial $h_{\delta,A'}(\gamma_2)_B$, which is linear in each block of $(A \setminus A') \sqcup B = S \setminus A'$. The set $A$ is nonempty because $|A| \geq |S|/2 > 0$.

Now take $\gamma$ to be the output gate and $S = [k]$, for which $3k > 2k$. Then $\gamma_S = f$, and grouping the terms by $(\delta, A)$ gives, for each balanced set $A$, at most one term for each of the $L$ gates $\delta$, with $g_{A,\delta} = \delta_A$ and $h_{A,\delta} = h_{\delta,A}$. □

**Corollary 15.3** (Circuit-Induced Decomposition of the Complete Network)**.** *If $f_{k,\ell}$ has a circuit of size $L$ over $\mathbb{F}$, then $T_{k,\ell} = \sum_{S \in \mathcal{B}_k} t_S$, where each $t_S$ is a sum of at most $L$ products of a tensor on $\tilde{S}$ and a tensor on the complementary ends, so that $r_{\tilde{S}}(t_S) \leq L$.*

*Proof.* The polynomial $f_{k,\ell}$ is linear in each block, so it equals its part of degree one in every block. Apply Proposition 15.2. The coefficient tensor of $g_{S,j} h_{S,j}$ is the tensor product of the coefficient tensor of $g_{S,j}$, indexed by the local views of the nodes of $S$, and that of $h_{S,j}$, indexed by the local views of the other nodes, because the monomials of the product are the products of the monomials of the factors. Splitting a local view into the $k-1$ strings of its edges is an isomorphism of index spaces, so these are tensors on $\tilde{S}$ and on its complement. □

**Definition 15.4** (Hub Network of a Balanced Set)**.** For $S \in \mathcal{B}_k$ let $o_S, o'_S$ be two parties not among the ends, with different parties for different sets. Define

$$T_S = \prod_{a \in S} W_{2^\ell}((a,b), o_S) \cdot W_L(o_S, o'_S) \cdot \prod_{a \notin S} W_{2^\ell}((a,b), o'_S),$$

where the products run over all ends $(a,b)$ with the stated condition on $a$.

**Proposition 15.5** (Simultaneous Spectral Inequality)**.** *Suppose $f_{k,\ell}$ has a circuit of size $L$. Let $V$ be the set of ends together with all hubs $o_S, o'_S$, and let $\varphi$ be a spectral point on $V$*

*with exponent pseudometric $d = d_\varphi$. Then there is $S \in \mathcal{B}_k$ with*

$$\ell \sum_{a<b} d\big((a,b),(b,a)\big) \le k + \ell \sum_{(a,b)} d\big((a,b), h_S(a,b)\big) + d(o_S, o'_S) \log L,$$

*where $h_S(a,b) = o_S$ if $a \in S$ and $h_S(a,b) = o'_S$ otherwise, and the sum is over all ends. Consequently, for every $c \ge 0$ the same inequality holds with $d$ replaced by $c\,d$ and with $k$ replaced by $ck$.*

*Proof.* Regard $T_{k,\ell}$ and the $t_S$ as tensors on $V$ with index set of one element at the hubs. By Corollary 15.3 and Proposition 10.3, $T_{k,\ell} = \sum_S t_S$ with $t_S \preceq T_S$, where the application of the proposition with the hubs $o_S, o'_S$ and with $r = L$, padding with zero terms when fewer than $L$ products occur, gives a restriction that acts only at the hubs and leaves the ends and the other hubs unchanged. The cover axiom gives $\varphi(T_{k,\ell}) \le \sum_S \varphi(T_S) \le 2^k \max_S \varphi(T_S)$. Take logarithms. By the additive length identity the left side is $\ell \sum_{a<b} d((a,b),(b,a))$ and, for the set $S$ attaining the maximum, $\log \varphi(T_S)$ is the sum on the right apart from $k$. Multiplying by $c \ge 0$ gives the scaled form. □

*Remark* 15.6. The statement in the drafting spine for this chapter used the distances between nodes and the factor $\ell(k-1)$ for the attachment of a node to a hub. The ends are the parties between which the wires run, and the formulation above, in which each end is placed separately, is the one that the geometry of Chapter 16 requires. Moreover the term $k$ is multiplied by the scale $c$ in the Euclidean comparison, and that factor is the source of the unspecified threshold in Proposition 16.4.

# Chapter 16

# Euclidean Configurations and the Lower-Bound Exponent

## 16.1 Placement of Wire Endpoints

The inequality of Chapter 15 holds for every spectral point on the set of ends and hubs, and its usefulness depends on choosing a point for which the distances are known. The flat hypothesis provides this: every finite configuration of points in a Euclidean space is approximated, after a common scaling, by the exponent pseudometric of a spectral point. The task becomes a problem of Euclidean geometry. The ends and hubs are to be placed in a real vector space so that the inequality of Chapter 15 fails for every possible choice of balanced set unless the circuit is large.

The first requirement is that the two ends of every edge lie at distance one, so that the length of the network is $\ell N$ in Euclidean terms. A configuration that achieves this with the largest possible symmetry places the end $(a, b)$ at one half of the unit vector $u_{ab} = (e_a - e_b)/\sqrt{2}$ in $\mathbb{R}^k$, where $e_1, \ldots, e_k$ is the standard basis. Then the end $(b, a)$ lies at the antipodal point, the two ends of the edge $ab$ are at distance $\|u_{ab}\| = 1$, and the positions of all ends add up to zero. The ends of the edges at the node $a$ lie in the half-spaces determined by the sign of the $a$th coordinate, and the sum of the positions of the ends at the nodes of a set $S$ is determined by the edges leaving $S$, because the contributions of edges with both ends at the nodes of $S$ cancel in pairs. This is the geometrical expression of the fact that only the edges crossing a cut matter.

## 16.2 Placement of Auxiliary Hubs

For a balanced set $S$ let $w_S$ be the sum of the vectors $u_{ab}$ over the pairs with $a \in S$ and $b \notin S$. The ends at the nodes of $S$ have positions whose sum is $\frac{1}{2}w_S$, and the other ends have positions whose sum is $-\frac{1}{2}w_S$. The hubs are placed at $\pm\eta\, w_S/\|w_S\|$ for a suitable length $\eta$, the hub $o_S$ in the direction of the ends it serves and the hub $o'_S$ opposite. A unit vector at the end of a displacement of length one half from the origin is closer to a hub in

its own direction than to the origin by an amount proportional to the inner product, and this saving, summed over the ends of $S$ and over the other ends, adds up to a quantity proportional to $\eta \|w_S\|$. The norm of $w_S$ is of order $k^{3/2}$ for balanced sets, because the number of crossing edges is of order $k^2$ and the vectors have unit length and add up along a common direction in as many as $k$ coordinates, and the choice of $\eta$ of order $k^{-1/2}$ makes the saving of order $k$ while the price of the hub displacement, quadratic in $\eta$, is smaller. The hubs of different balanced sets are different parties even when their positions coincide, since the wires attached to them belong to different tensors.

## 16.3 Geometric Savings

The saving per balanced set is the amount by which the total length from the ends to the hubs falls short of the network length $N$, and it is of order $k$. After scaling by $\ell$ it becomes of order $\ell k$, and this quantity must be paid, in the inequality of Chapter 15, by the term $d(o_S, o'_S) \log L$ in which the hub separation is of order $k^{-1/2}$. The surplus is therefore divided by a small distance, which is what converts a saving of order $\ell k$ into a lower bound of order $\ell k^{3/2}$ on $\log L$. Two terms work against this. The cover contributes $k$ bits multiplied by the scale $c$ of the approximation, and $c$ is unknown, depending on $k$ through the choice of spectral point. This forces the length $\ell$ to exceed a threshold $\ell_0$ proportional to $c$ in the first bound, and it is the reason that the first bound is only an existence statement about thresholds. The errors of the approximation of Euclidean distances by scaled exponents contribute terms of order $k^2 \ell \delta$ that are kept below the saving by choosing the tolerance $\delta$ of order $1/k$.

## 16.4 The Resulting Lower Bound

The conclusion of the first argument is that, in the flat case and for $\ell \geq \ell_0(k)$, every circuit for the network polynomial has at least $2^{\ell k^{3/2}/6}$ gates. Together with the reduction of the network polynomial to the permanent this excludes polynomial size circuits for the permanent: for a given exponent $C$ one fixes $k$ large enough that $\sqrt{k} \geq 30(C+1)$, takes $\ell$ as large as the permanent of the given size allows, and finds that the bound exceeds the size $m^C$ at the corresponding scale. This is the first conditional theorem of substance in the monograph, and it is conditional on the flat hypothesis and on the inherited reduction of Chapter 14.

The refinement that gives an explicit threshold replaces the cover term. When the $J$ terms of a cover arise by applying linear functionals at parties outside a set $P$, as is the case for the hub restrictions, which act only at the hubs, the factor $J$ in the cover inequality can be replaced by $J^{g(P)}$, where $g(P)$ is the exponent of the unit tensor on $P$, and the scaled exponent $c\, g(P)$ is a distance-like quantity that can be controlled geometrically. If the ends are placed at the centres of the facets of a cube, a set $P$ of such points has scaled exponent at most about $\sqrt{k}$ by an induction on the dimension in which each step passes to midpoints,

and the cover term becomes $k$ times about $\sqrt{k}$, that is of order $k^{3/2}$, which is smaller than the saving of order $\ell k$ as soon as $\ell \geq 100\sqrt{k}$. Both ingredients, the replacement of the cover and the bound for facet centres, rest on the nonpositive curvature of Chapter 12 and on an averaging construction of spectral points, and they are inherited in the formal statements below. The explicit conclusion is $\log L \geq \ell k^{3/2}/14$ for $\ell \geq 100\sqrt{k}$.

## 16.5 Parameter Optimization

The permanent of a matrix of size $m$ gives a circuit for the network polynomial with parameters $k$ and $\ell$ as long as $m \geq 3k^3 \ell\, 2^{k\ell}$, and the lower bound for the network polynomial then bounds the circuit size of the permanent. The bound grows with $\ell$ for fixed $k$, while $\log m$ grows like $k\ell$, so one makes $\ell$ as small as the theorem permits, namely $\ell = 100\sqrt{k}$. Writing $k = r^2$ and $\ell = 100r$ gives $\ell k^{3/2} = 100r^4$ and $k\ell = 100r^3$, so that $\log m$ is of order $r^3$ and the bound $2^{\Omega(r^4)}$ becomes

$$2^{\Omega((\log m)^{4/3})} = m^{\Omega((\log m)^{1/3})}.$$

The exponent $4/3$ is therefore the quotient of two scales that appear in the argument. The obstruction scale is $\ell k^{3/2}$, arising as the product of the length $\ell$ of the encoding, the number $k$ of nodes, and the factor $\sqrt{k}$ that comes from dividing the saving of order $k$ by the hub separation of order $k^{-1/2}$. The encoding scale is $k\ell$, the logarithm of the size of the matrix needed to represent the network polynomial as a permanent. With $\ell \sim \sqrt{k}$, the obstruction scale is $k^2$ and the encoding scale is $k^{3/2}$, and the exponent is the ratio of the logarithms, $2/(3/2) = 4/3$. It emerges from the interaction between the geometry and the representation, and not from counting monomials, edges, or assignments.

## 16.6 Possible Improvements

The threshold $\ell \geq 100\sqrt{k}$ is forced by the cover term of order $k^{3/2}$ against a saving of order $\ell k$, and any improvement of the exponent would have to reduce this term, either by a better bound for the exponent of the set of ends or by a placement of the ends that makes it smaller, or else would have to increase the saving per balanced set beyond order $k$. The constants of the argument are not optimized, and the exponent $4/3$ is the one that this configuration produces. The relation between this configuration and the optimal one is not known, and nothing in the present text suggests that the exponent is best possible for the method. These remarks describe the structure of the argument and are not claims about what a different configuration could achieve.

## 16.7 Formal Statements

**Definition 16.1** (Euclidean Placement)**.** Let $e_1, \dots, e_k$ be the standard basis of $\mathbb{R}^k$ and $u_{ab} = (e_a - e_b)/\sqrt{2}$, so that $u_{ba} = -u_{ab}$ and $\|u_{ab}\| = 1$. Place the end $(a, b)$ at $\frac{1}{2}u_{ab}$. For

$S \in \mathcal{B}_k$ let $w_S = \sum_{a \in S,\, b \notin S} u_{ab}$ and place the hubs at $o_S = \eta\, w_S / \|w_S\|$ and $o'_S = -o_S$ with $\eta = \dfrac{1}{6\sqrt{k}}$.

**Proposition 16.2** (Euclidean Estimates)**.** *In the placement of Definition 16.1, the two ends of every edge are at distance* 1*, and for every $S \in \mathcal{B}_k$ one has $\|w_S\|^2 = \frac{1}{2} k\,|S|(k-|S|) \geq k^3/9$, $\|o_S - o'_S\| = \dfrac{1}{3\sqrt{k}}$, and*

$$\sum_{(a,b)} \|\tfrac{1}{2} u_{ab} - h_S(a,b)\| \leq N - \frac{k}{12},$$

*where $h_S(a,b) = o_S$ for $a \in S$ and $h_S(a,b) = o'_S$ otherwise, and the sum is over all $k(k-1)$ ends.*

*Proof.* The ends $(a,b)$ and $(b,a)$ sit at $\pm\frac{1}{2}u_{ab}$, at distance $\|u_{ab}\| = 1$. Write $s = |S|$. Then $w_S = \frac{1}{\sqrt{2}}\big[(k-s)\sum_{a \in S} e_a - s \sum_{b \notin S} e_b\big]$, whence $\|w_S\|^2 = \frac{1}{2}\big[(k-s)^2 s + s^2(k-s)\big] = \frac{1}{2} k\, s(k-s)$, and $s(k-s) \geq \frac{2}{9}k^2$ by the proof of Proposition 14.3 gives $\|w_S\|^2 \geq k^3/9$. The distance between the hubs is $2\eta = 1/(3\sqrt{k})$.

Since $u_{ab} + u_{ba} = 0$, the vectors $u_{ab}$ with $a \in S$ and $b \neq a$ add up to $w_S$, and those with $a \notin S$ add up to $-w_S$. For a unit vector $u$ and a vector $o$ one has $\|\frac{1}{2}u - o\|^2 = \frac{1}{4} - u \cdot o + \|o\|^2$, and $\sqrt{\frac{1}{4} + x} \leq \frac{1}{2} + x$ for $x \geq -\frac{1}{4}$, because $(\frac{1}{2} + x)^2 \geq \frac{1}{4} + x$. Here $x = -u \cdot o + \eta^2 \geq -\eta \geq -\frac{1}{4}$. Hence $\|\frac{1}{2}u - o\| \leq \frac{1}{2} - u \cdot o + \eta^2$. Summing over the ends at the nodes of $S$ with $o = o_S$, and over the others with $o = -o_S$, the total is at most

$$\frac{k(k-1)}{2} - w_S \cdot o_S - w_S \cdot o_S + k(k-1)\eta^2 \leq N - 2\eta\|w_S\| + k^2\eta^2,$$

using $w_S \cdot o_S = \eta\|w_S\|$ and $\frac{1}{2}k(k-1) = N$. With $\|w_S\| \geq k^{3/2}/3$ and $\eta = 1/(6\sqrt{k})$ one has $2\eta\|w_S\| \geq k/9$ and $k^2\eta^2 = k/36$, so the total is at most $N - k/9 + k/36 = N - k/12$. □

**Definition 16.3** (Flat Condition in Use)**.** Say that the flat condition holds over $\mathbb{F}$ if for every finite set $V \subset \mathbb{R}^n$ and $\delta > 0$ there are a spectral point $\varphi$ on $V$ and $c \geq 0$ with $|c\, d_\varphi(x,y) - \|x - y\|\,| < \delta$ for all $x, y \in V$, where $c$ and $\varphi$ may depend on $V$ and $\delta$. Pulling back along a map of parties onto $V$, which may identify the positions of distinct parties, yields a spectral point on the set of parties with the same property.

**Proposition 16.4** (First Lower Bound for the Network Polynomial)**.** *Assume the flat condition over $\mathbb{F}$. For every $k \geq 2$ there is $\ell_0$ such that for every $\ell \geq \ell_0$ every circuit over $\mathbb{F}$ for $f_{k,\ell}$ has at least $2^{\ell k^{3/2}/6}$ gates.*

*Proof.* Place the ends and hubs as in Definition 16.1, regarded as distinct parties for distinct balanced sets, and let $\delta = 1/(150k)$. By the flat condition there are a spectral point $\varphi$ on the set of these parties and $c \geq 0$, both depending on $k$ only, with $|c\, d_\varphi - \text{Euclidean distance}| < \delta$. Put $\ell_0 = \lceil 100c \rceil + 1$ and let $\ell \geq \ell_0$. Suppose $f_{k,\ell}$ has a circuit of size $L$. Proposition 15.5,

multiplied by $c$, gives a balanced set $S$ with

$$\ell \sum_{a<b} c\, d\big((a,b),(b,a)\big) \le ck + \ell \sum_{(a,b)} c\, d\big((a,b),h_S(a,b)\big) + c\, d(o_S,o_S')\log L.$$

Each edge term on the left exceeds $1-\delta$, so the left side is at least $\ell N(1-\delta)$. Each of the $2N$ end-to-hub terms exceeds its Euclidean value by less than $\delta$, so by Proposition 16.2 the sum on the right is at most $\ell(N - k/12 + 2N\delta)$, and the last term is at most $(2\eta+\delta)\log L$ with $2\eta = 1/(3\sqrt{k})$. Hence

$$\ell N(1-\delta) \le ck + \ell\Big(N - \frac{k}{12} + 2N\delta\Big) + (2\eta+\delta)\log L,$$

and therefore $\ell(\frac{k}{12} - 3N\delta) \le ck + (2\eta+\delta)\log L$. Now $ck \le \ell k/100$ by the choice of $\ell_0$, and $3N\delta \le \frac{3}{2}k^2 \cdot \frac{1}{150k} = k/100$. It follows that $\ell k(\frac{1}{12} - \frac{1}{50}) \le (2\eta+\delta)\log L$. For $k \ge 2$ one has $2\eta+\delta \le (\frac{1}{3} + \frac{1}{150\sqrt{2}})k^{-1/2} \le 0.34\, k^{-1/2}$, and $\frac{1}{12} - \frac{1}{50} = \frac{19}{300}$, so $\log L \ge \dfrac{19}{300 \cdot 0.34}\, \ell k^{3/2} > \ell k^{3/2}/6$. □

**Theorem 16.5** (No Polynomial Size Permanent Circuits in the Flat Case)**.** *Assume the flat condition over a field $\mathbb{F}$ with $2 \neq 0$, and Inherited Result 14.7. For every $C$ there is $m_0$ such that for $m \ge m_0$ no circuit over $\mathbb{F}$ with at most $m^C$ gates computes* $\mathrm{PERM}_m$.

*Proof.* Fix $k$ with $\sqrt{k} \ge 30(C+1)$. For large $m$ let $\ell$ be the largest integer with $3k^3\ell\, 2^{k\ell} \le m$. Then $m < 3k^3(\ell+1)2^{k(\ell+1)}$, so $\log m \le 2k\ell$ once $\ell$ is large, and $\ell \to \infty$ with $m$, so $\ell \ge \ell_0$ for large $m$. A circuit with at most $m^C$ gates for $\mathrm{PERM}_m$ gives, by Inherited Result 14.7, a circuit for $f_{k,\ell}$ with at most $5m^C$ gates, whence $\ell k^{3/2}/6 < \log(5m^C) \le 3 + 2Ck\ell$ by Proposition 16.4. But $k^{3/2}/6 \ge 5(C+1)k$, so the left side exceeds $5(C+1)k\ell > 3 + 2Ck\ell$, a contradiction. □

**Inherited Result 16.6** (Sums With Few Terms)**.** *Let $P$ be a nonempty set of parties, $t$ a tensor on $P$, and $T_1, \dots, T_J$ tensors with the same index sets as $t$ at the parties of $P$ such that $t = \sum_j f_j(T_j)$, each $f_j$ consisting of one linear functional at every party outside $P$. Then $\varphi(t) \le J^{g_\varphi(P)} \max_j \varphi(T_j)$ for every spectral point $\varphi$, where $g_\varphi(P)$ is the exponent of the unit tensor on $P$. This is Lemma E.7, proved by expanding a high power of $t$ by words, grouping words by type, and letting the power tend to infinity.*

**Inherited Result 16.7** (Facet Centres of a Cube)**.** *In the flat case let $V \subset \mathbb{R}^k$ be finite, $P \subseteq V \cap \{\pm\frac{1}{2}e_1, \dots, \pm\frac{1}{2}e_k\}$ nonempty and $\varepsilon > 0$. There are a spectral point $\varphi$ on $V$ and $c \ge 0$ with $|c\, d_\varphi(x,y) - \|x-y\|\,| < \varepsilon$ on $V$ and $c\, g_\varphi(P) \le \sqrt{k} + \varepsilon$. This is Proposition E.9, obtained by passing $k$ times to midpoints from the two ends of a main diagonal using Lemma E.8, which is a consequence of Theorem C.1 and of the averaging construction of Proposition A.7.*

**Proposition 16.8** (Explicit Lower Bound for the Network Polynomial)**.** *Assume the flat condition and the two preceding inherited results. For $k \ge 2$ and $\ell \ge 100\sqrt{k}$ every circuit for $f_{k,\ell}$ has at least $2^{\ell k^{3/2}/14}$ gates, and in particular more than $2^{\ell k^{3/2}/16}$.*

*Proof.* Place the ends at facet centres of the cube $[-\frac{1}{2},\frac{1}{2}]^k$ by putting $(a,b)$ at $\frac{1}{2}e_a$ and $(b,a)$ at $-\frac{1}{2}e_a$ for $a<b$, so that the two ends of every edge are at distance 1 and all positions add up to zero. For a balanced $S$ let $w_S$ be twice the sum of the positions of the ends $(a,b)$ with $a\in S$. If $c_a$ is the number of $b>a$ such that exactly one of $a,b$ lies in $S$, then $w_S=\sum_a \pm c_a e_a$ and $\sum_a c_a = |S|(k-|S|)\geq 2k^2/9$, so by the Cauchy–Schwarz inequality $\|w_S\|^2\geq(\sum c_a)^2/k\geq 4k^3/81$ and $\|w_S\|\geq\frac{2}{9}k^{3/2}$. Place the hubs at $\pm\eta\, w_S/\|w_S\|$ with $\eta=\dfrac{2}{9\sqrt{k}}$. For $\|r\|=\frac{1}{2}$ and $\|o\|=\eta$ one has $\|r-o\|^2=\frac{1}{4}-2r\cdot o+\eta^2$ and hence $\|r-o\|\leq\frac{1}{2}-2r\cdot o+\eta^2$, and summing over the ends as in Proposition 16.2 the distances from the ends to their hubs add up to at most $N-2\eta\|w_S\|+k^2\eta^2\leq N-\frac{4}{81}k$.

Let $\delta=10^{-3}/k$. By Inherited Result (Facet Centres) with $P$ the set of positions of the ends, there are $\varphi$ and $c$ with $c\,d_\varphi$ within $\delta$ of the Euclidean distances on the positions of all ends and hubs and $c\,g_\varphi(P)\leq\sqrt{k}+\delta$; pulling back to the ends as distinct parties, the exponent of the set $\mathcal{E}$ of all ends satisfies $c\,g_\varphi(\mathcal{E})\leq\sqrt{k}+\delta$, since the push-forward of the unit tensor on $\mathcal{E}$ is that on $P$. As in the proof of Proposition 16.4 one has $T_{k,\ell}=\sum_S t_S$ with $t_S\preceq T_S$, the restriction applying a functional at $o_S$ and one at $o'_S$ and nothing at the ends, and there are at most $2^k$ terms. The inherited lemma on sums with few terms, with $P=\mathcal{E}$ and $J\leq 2^k$, gives $\varphi(T_{k,\ell})\leq 2^{k\,g_\varphi(\mathcal{E})}\max_S\varphi(T_S)$. Taking logarithms and multiplying by $c$,

$$\ell N(1-\delta)\leq k(\sqrt{k}+\delta)+\ell\Big(N-\frac{4k}{81}+\delta k^2\Big)+(2\eta+\delta)\log L$$

for some $S$. Since $\ell\geq 100\sqrt{k}$, one has $k(\sqrt{k}+\delta)\leq 1.5k^{3/2}\leq 0.015\,\ell k$, and the two terms with $\delta$ contribute at most $\frac{3}{2}\delta k^2\ell=0.0015\,\ell k$. Hence $(2\eta+\delta)\log L\geq\ell k(\frac{4}{81}-0.015-0.0015)\geq\ell k/31$. As $2\eta+\delta\leq 0.446\,k^{-1/2}$, it follows that $\log L\geq\ell k^{3/2}/14$. □

**Theorem 16.9** (Flat-Case Lower Bound for the Permanent)**.** *Assume the flat condition over a field with $2\neq 0$, and the inherited results of Chapters 14 and 16. There is $m_0$ such that for $m\geq m_0$ every circuit for* $\mathrm{PERM}_m$ *has more than $2^{(\log m/1600)^{4/3}}$ gates.*

*Proof.* For large $m$ let $r$ be the largest integer with $200r^3\leq\lfloor\log m\rfloor$, and put $k=r^2$ and $\ell=100r$, so that $\ell=100\sqrt{k}$. For large $m$ one has $r\geq 2$, $3k^3\ell\,2^{k\ell}=300r^7\,2^{100r^3}\leq 2^{200r^3}\leq m$, and $\log m\leq 1600r^3$, because maximality of $r$ gives $\log m<200(r+1)^3+1\leq 1600r^3$. By Inherited Result 14.7 and Proposition 16.8, a circuit with $L$ gates for $\mathrm{PERM}_m$ gives $\log(5L)\geq\ell k^{3/2}/16=\frac{25}{4}r^4$. Hence $\log L>r^4$ for $r\geq 2$, and $r^3\geq\log m/1600$ gives $r^4\geq(\log m/1600)^{4/3}$. □

**Proposition 16.10** (Origin of the Exponent)**.** *In the parametrization $k=r^2$, $\ell=100r$ one has $\log m=\Theta(r^3)$ and $\ell k^{3/2}=\Theta(r^4)$, so that $2^{\Omega(\ell k^{3/2})}=2^{\Omega((\log m)^{4/3})}=m^{\Omega((\log m)^{1/3})}$.*

*Proof.* By the proof of Theorem 16.9, $200r^3\leq\log m+1$ and $\log m\leq 1600r^3$, so $\log m=\Theta(r^3)$ and $r^4=\Theta((\log m)^{4/3})$. Moreover $2^{c(\log m)^{4/3}}=m^{c(\log m)^{1/3}}$. □

**Inherited Result 16.11** (Transfer to the Hamiltonian Cycle Polynomial and to the Field of the Dichotomy)**.** *The conclusion of Theorem 16.9 holds for* $\mathrm{HC}_m$ *in every characteristic,*

*with* 3 *replaced by* 40 *in the condition on the matrix size, and circuits over an arbitrary field are reduced to circuits over the field over which the flat or contracting alternative of Chapter 13 is established. These are Proposition M.2 and Lemma M.1 of the manuscript.*

**Proof Obligation 16.12** (Flat-Case Reconstruction)**.** *The conditional statement* $\mathsf{F}(\mathbb{F}) \Rightarrow L_{\mathbb{F}}(\mathrm{HC}_m) \geq 2^{\Omega((\log m)^{4/3})}$ *follows from Theorem 16.9 once the following inherited ingredients have been reconstructed: the permanent and Hamiltonian cycle realizations of clause sums (Proposition E.1, Corollary E.2, Proposition M.2), the spectral exponents and cover axiom (Lemma A.2, Definition A.1), the sums-with-few-terms lemma (Lemma E.7), the midpoint convexity of the exponent (Lemma E.8 with Proposition A.7 and Theorem C.1), and the facet-centres proposition (Proposition E.9). The constants have been recomputed here for the permanent and agree with those of the source up to the choice of the absolute constant in the exponent.*

# Part VII

# The Contracting-Case Lower Bound

# Chapter 17

# Contraction and Poincaré Inequalities

## 17.1 Unitriangular Matrix Groups

The contracting case concerns a hypothesis about prices on matrix spaces over the field with two elements, and the first task is to convert it into a statement about a family of finite groups whose Cayley graphs can serve as skeletons for tensor networks. The groups are the upper unitriangular groups $U_h$ of $h \times h$ matrices over $\mathbb{F}_2$ with ones on the diagonal and zeros below it. They are chosen for three properties. They are large, since $|U_h| = 2^{h(h-1)/2}$, so that a single element carries a number of bits of order $h^2$. They are generated by the $h - 1$ elementary matrices $1 + E_{j,j+1}$ that join consecutive coordinates, so that the Cayley graph has small degree and each generator affects only a local pair of coordinates. And they contain, as additive subgroups, the matrix spaces $\mathrm{Mat}_\mu(\mathbb{F}_2)$ that appear in the contracting hypothesis, for instance as the off-diagonal block of a unitriangular matrix of size $2\mu$ [30]. The last property is what allows the strict contraction assumed for rank displacement on matrix spaces to be transported into an inequality on the groups.

## 17.2 Poincaré Inequalities

A Poincaré inequality bounds the global variation of a function by its local variation. For a finite group with a generating set, the global variation of a map into a metric space is the mean squared distance between the images of two random elements, and the local variation is the sum over the generators of the mean squared distance between the images of an element and its neighbour. The inequality asserts that the first is at most a constant times the second. For maps into a Euclidean space the inequality is a spectral gap statement for the Cayley graph, and the constant is governed by the gap. The theorem used here is a version for maps into price spaces, which are nonpositively curved by Chapter 12, and it holds with a constant that is independent of $h$ once the contracting hypothesis is available [30].

The mechanism of the proof, which is inherited here, uses the action of $U_h$ on the space of maps from $U_h$ into a CAT(0) space and the geometry of fixed points. Subgroups isomorphic to $U_{2u}$, generated by the generators in overlapping windows of consecutive coordinates, have fixed point sets in this space. The contracting hypothesis, through the length functions of Section 17.4, forces these fixed point sets to meet almost at right angles, and a map is moved toward a common fixed point by averaging its projections onto the window fixed sets. The sum of the squared distances to the fixed sets decreases by a fixed factor at each averaging, and the existence of a map with all windows fixed, which is a constant map, bounds the variation by the energy. The theorem is a delicate piece of geometry and is the first of the results in the contracting case on which the later reconstruction depends.

## 17.3 Uniformity Across Scales

The constant in the Poincaré inequality does not depend on $h$ when $h$ ranges over the multiples $lu$ of a fixed integer $u$ with $l \geq 2$, and this uniformity is what makes the group $U_h$ with a single large $h$ usable throughout the rest of the construction. A constant that grew with $h$ would be absorbed by the choice of the parameters in the rerouting laws of the next chapter only at the cost of making these parameters depend on one another circularly. The source fixes $u$ and the constant once and for all, and then fixes a single multiple $h$ of $u$ so large that two numerical inequalities, which involve the constant, are satisfied. All later constants depend only on these choices. The uniformity across scales is therefore a precondition for an argument in which the same laws are applied at every stage of a tower of groups of unbounded size.

## 17.4 Length Functions and Contraction

A length function on a group assigns to each element a nonnegative number with $|1| = 0$, $|g^{-1}| = |g|$ and $|gg'| \leq |g| + |g'|$. It contracts if, for every homomorphism $\theta$ from the additive group $\mathrm{Mat}_\mu(\mathbb{F}_2)$ into the group, the mean of $|\theta W|^2$ over the matrices of rank $\mu$ is at most $s^2$ times the mean over the matrices of rank one. The relevance is that every map from a finite group into a price space induces such a length function, so that the hypothesis on prices becomes a property of the group-theoretic quantity that the Poincaré inequality controls, and the translation is exact in the formal statements below.

## 17.5 Formal Statements

**Definition 17.1** (Unitriangular Group)**.** For $h \geq 2$ let $U_h$ be the group of $h \times h$ matrices over $\mathbb{F}_2$ with $A_{ii} = 1$ and $A_{ij} = 0$ for $i > j$, under matrix multiplication. Number the coordinates $0, \ldots, h-1$, and let $x_j = 1 + E_{j,j+1}$ for $0 \leq j < h-1$, where $E_{jk}$ is the matrix whose only nonzero entry is a 1 in row $j$ and column $k$.

**Proposition 17.2** (Cardinality and Generation)**.** *The group $U_h$ has order $2^{h(h-1)/2}$, each $x_j$ is an involution, $x_i$ and $x_j$ commute when $|i-j| \geq 2$, and the elements $x_0, \dots, x_{h-2}$ generate $U_h$.*

*Proof.* A matrix above the diagonal can be chosen freely, which gives $2^{h(h-1)/2}$ matrices, and the set is closed under products and inverses because the product of two upper triangular matrices with unit diagonal is again of this kind. Since $E_{j,j+1}^2 = 0$ we have $x_j^2 = 1+2E_{j,j+1} = 1$ in characteristic two. If $|i-j| \geq 2$ then $E_{i,i+1}E_{j,j+1} = E_{j,j+1}E_{i,i+1} = 0$, so the generators commute. For generation, write $e_{ik} = 1 + E_{ik}$ with $i < k$. For $i < j < k$ the commutator $e_{ij}e_{jk}e_{ij}^{-1}e_{jk}^{-1}$ equals $e_{ik}$, because $E_{ij}E_{jk} = E_{ik}$ and all other products of the matrix units occurring vanish, so by induction on $k - i$ every $e_{ik}$ lies in the group generated by the $x_j = e_{j,j+1}$. Every element of $U_h$ is a product of elements $e_{ik}$ by column elimination. □

**Definition 17.3** (Variance and Energy of a Map)**.** For a finite set $Y$ with a price $d$ and a map $f : U_h \to Y$, write $|a, a'| = d(f(a), f(a'))$. The variance of $f$ is $\mathcal{V}_f = \mathbb{E}_{g,a \in U_h} |a, ga|^2$ and its generator energy is $\mathcal{E}_f = \sum_{j<h-1} \mathbb{E}_{a \in U_h} |a, x_j a|^2$.

The generators act by left multiplication here, which is the convention of Theorem F.6 of the source. For a price that is not invariant under the group the quantities for left and right multiplication differ, and the drafting spine for this chapter, which wrote $d(g, gs_i)$, should be read with the convention above.

**Proposition 17.4** (Length Functions from Prices)**.** *Assume the contracting condition with parameters $\mu$ and $s$. Let $G$ be a finite group, $d$ a price on a set $Y$ and $f : G \to Y$ a map. Then $|g| = \left(\mathbb{E}_{a \in G}\, d(f(a), f(ga))^2\right)^{1/2}$ is a length function that contracts: for every homomorphism $\theta : \mathrm{Mat}_\mu(\mathbb{F}_2) \to G$, $\mathbb{E}_{\mathrm{rank}\, W = \mu} |\theta W|^2 \leq s^2\, \mathbb{E}_{\mathrm{rank}\, W = 1} |\theta W|^2$.*

*Proof.* That $|1| = 0$ and $|g^{-1}| = |g|$ is immediate, the latter by the substitution $a \mapsto g^{-1}a$. For the triangle inequality, $|gg'| = \left(\mathbb{E}_a d(f(a), f(gg'a))^2\right)^{1/2} \leq \left(\mathbb{E}_a d(f(a), f(g'a))^2\right)^{1/2} + \left(\mathbb{E}_a d(f(g'a), f(gg'a))^2\right)^{1/2}$ by the triangle inequality for $d$ and for the $L^2$ norm, and the second term equals $|g|$ by the substitution $a \mapsto g'^{-1}a$. For a homomorphism $\theta$ and $a_0 \in G$ define $d_{a_0}(W_1, W_2) = d\big(f(\theta(W_1)a_0), f(\theta(W_2)a_0)\big)$ on $\mathrm{Mat}_\mu(\mathbb{F}_2)$. This is the pullback of a price along a map, hence a price. Since $\theta$ is a homomorphism, $\theta(W_1 + W) = \theta(W)\theta(W_1)$ in the group $G$, and the mean displacement of this price is $D_{d_{a_0}}(r)^2 = \mathbb{E}_{W_1} \mathbb{E}_{\mathrm{rank}\, W = r} d\big(f(\theta(W_1)a_0), f(\theta(W)\theta(W_1)a_0)\big)^2$. For fixed $W_1$ the element $\theta(W_1)a_0$ is uniform in $G$ when $a_0$ is, so $\mathbb{E}_{a_0} D_{d_{a_0}}(r)^2 = \mathbb{E}_{\mathrm{rank}\, W = r} |\theta W|^2$. The contracting condition gives $D_{d_{a_0}}(\mu)^2 \leq s^2 D_{d_{a_0}}(1)^2$ for every $a_0$, and averaging over $a_0$ gives the claim. □

**Inherited Result 17.5** (Uniform Poincaré Inequality)**.** *Assume the contracting condition. There are an integer $u \geq 2$ and a real $C_0 \geq 0$ such that for every $l \geq 2$, with $h = lu$, and every price on the set $U_h$,*

$$\mathbb{E}_{g,a \in U_h} |a, ga|^2 \leq C_0 \sum_{j<h-1} \mathbb{E}_{a \in U_h} |a, x_j a|^2.$$

*This is Theorem F.6 of the manuscript, resting on Lemmas F.1 to F.5, the length-function remark of Section F.1, and the CAT(0) realization of prices.*

**Corollary 17.6** (Edge Form of the Inequality)**.** *Let $b = |U_h|$ with $h$ as in Inherited Result 17.5, let $C_0' \geq 2C_0 + 1$ be an integer, and write $|e|$ for the distance between the two elements of a pair $e$. For every price on $U_h$,*

$$\frac{2}{b(b-1)} \sum_{e} |e|^2 \leq C_0' \cdot \frac{2}{b} \sum_{edges\ e} |e|^2,$$

*where the left sum runs over all pairs of distinct elements and the right sum over the edges $\{a, x_j a\}$ of the Cayley graph, of which there are $b/2$ for each $j$.*

*Proof.* The left side of the Poincaré inequality equals $b^{-2} \sum_{(a,a')} |a, a'|^2 = \frac{b-1}{b} \cdot \frac{2}{b(b-1)} \sum_e |e|^2 \geq \frac{1}{2} \cdot \frac{2}{b(b-1)} \sum_e |e|^2$ for $b \geq 2$, the sum over ordered pairs counting each unordered pair twice and the pairs with $a = a'$ contributing zero. Since each $x_j$ is an involution, $\sum_a |a, x_j a|^2 = 2 \sum_{\text{edges of type } j} |e|^2$, so the right side is $C_0 \cdot \frac{2}{b} \sum_{\text{edges}} |e|^2$. Combining, $\frac{2}{b(b-1)} \sum_e |e|^2 \leq 2C_0 \cdot \frac{2}{b} \sum_{\text{edges}} |e|^2 \leq C_0' \cdot \frac{2}{b} \sum_{\text{edges}} |e|^2$. □

**Definition 17.7** (Choice of the Scale)**.** Let $u$ and $C_0$ be as in Inherited Result 17.5, $C_0' \geq 2C_0 + 1$ an integer and $C_1 = 4C_0'$. Fix $h$, a multiple of $u$ with $h \geq 2u$, so large that $h \geq 3$ and $1 + 8C_1 \leq h/4$, and put $b = |U_h| \geq 8$. All constants of Chapters 18 to 21 depend only on these choices.

# Chapter 18

# Duality and Rerouting Identities

## 18.1 From Inequalities to Identities

The Poincaré inequality is a statement about prices, which are limits of exponent pseudometrics of spectral points. The duality theorem of Chapter 8 converts statements about all spectral points into statements about covers by tensors. The combination is the central device of the contracting case: an inequality that holds for every spectral point yields, through duality, an identity or a cover of tensors, and in this case the cover is a recipe for replacing wires between arbitrary pairs of elements of $U_h$ by wires along the edges of its Cayley graph. A wire between distant elements is expensive in a geometry in which only neighbours are close, and the Poincaré inequality says that the cost of the wires between all pairs is at most a constant times the cost of the wires along the edges. Duality turns this comparison of costs into a comparison of tensors, in which many copies of the all-pairs network are written as a short sum of networks that use only edges.

The elementary half of this passage is stated for later use. If a tensor $s$ has the property that every power $s^{\otimes n}$ is covered by $N$ restrictions of $t^{\otimes n}$, then every spectral point satisfies $\varphi(s)^n \leq N\varphi(t)^n$, and consequently $\varphi(s)/\varphi(t) \leq N^{1/n}$. The deep half is the converse, which guarantees covers once the inequality $\varphi(s) \leq \max_\beta \varphi(t_\beta)$ holds for all spectral points. That converse is the duality theorem and is inherited.

## 18.2 Rerouting Laws

A law is a recipe that replaces a bundle of wires between arbitrary pairs of elements by a sum of networks in which every wire runs along an edge. It consists of slots, which are the wires to be replaced and are counted by a load on pairs, a finite set of indices that enumerate the terms of the sum and carry a code of a fixed number of bits, channels, which are the edge wires of the new networks and have lengths that depend on the index, and node tensors, which say what each element does with its slots and its channels. The defining property is that the product of equality tests for all the slots equals the sum over indices and channel contents of the product of the node tensors. Because this is a finite family of

polynomial identities in the entries of the node tensors, it holds over every commutative algebra over the ring generated by those entries, which is a finitely generated ring. This last fact is what later allows the argument to specialize to a finite field.

The theorem on the existence of laws asserts that finitely many laws suffice to reroute every load on pairs that is not too large in a quadratic norm, and that the channels of every law have loads that are at most a constant times the square root of the scale in the dual norm. The proof applies the duality theorem to the inequality between spectral points that follows from the Poincaré inequality by two applications of the Cauchy–Schwarz inequality, and reads the resulting cover as a law. The relation between the numbers in the theorem and the constant of the Poincaré inequality is the sense in which the geometry of the contracting case enters the algebra of the rerouting.

## 18.3 Tree Decompositions

To apply the laws repeatedly the source uses a tower of finite groups with $h$ types of edges arranged cyclically, so that leaving out any one type leaves copies of the Cayley graph of $U_h$. The groups are the special linear groups over truncated polynomial rings, the tower is given by a chain of quotients with equivariant projections that are $b$ to one, and only the packets on the least loaded type of edge travel on to the next stage while all others end at a parent. Rerouting the former multiplies their load by a factor that does not depend on $h$, the quadratic load is halved at each stage by the choice of $h$, and the number of elements falls by the factor $b$. The result is a decomposition theorem: the value of any tensor network with $M$ nodes is a sum, over sequences of outcomes of the law moves, of the values of networks whose nodes form a rooted tree of depth polylogarithmic in $M$, in which the packets that pass through any one node number about $M^{-\gamma}$ times the packets in all, with $\gamma = 1/\log b$. The tree is where a computation will later be cut.

## 18.4 The Algebraic Cost of Rerouting

A representation of a network value as a sum of values of tree networks is not by itself an efficient decomposition, and the accounting of the decomposition is where the quantitative content lies. Each law move is a restructuring of the network that does not change the value, but it replaces one network by a sum over outcomes, and the number of terms, the number of nodes, the number of packets, and the sizes of the node tensors must all be controlled. The theorem states the bounds that hold: polylogarithmic depth, a bound on the number of packets at each node that is smaller by the factor $M^{\gamma}$ than the bound for all packets together, and the existence of a straight-line program of polynomial size that computes the structure of each tree from the sequence of outcomes without reference to the contents of the packets or the node tensors. This last property, that the structure depends only on numbers and not on tensors, is what allows the decomposition to be carried out by a program, and it is the property on which the cutting of computations in the next chapter rests. The proof of

the program's correctness is given in the source only by a description and an accounting of its size, and its exactness is established in the accompanying Lean development, so that this part of the argument has the status of a machine-verified statement whose printed proof is incomplete.

## 18.5 Formal Statements

**Definition 18.1** (Asymptotic Restriction Expenditure)**.** For tensors $s, t$ on a common set of parties and $n \geq 1$ let

$$R_n(s \mid t) = \inf\Big\{ N : \; s^{\otimes n} = \sum_{j=1}^{N} u_j, \; u_j \preceq t^{\otimes n} \Big\},$$

with $R_n(s \mid t) = \infty$ if there is no such $N$.

**Proposition 18.2** (Spectral Expenditure Bound)**.** *For every spectral point $\varphi$ and every $n \geq 1$, $\varphi(s)^n \leq R_n(s \mid t)\, \varphi(t)^n$. Hence $R_n(s \mid t)^{1/n} \geq \varphi(s)/\varphi(t)$ when $t \neq 0$ and $R_n(s \mid t) < \infty$, and*

$$\liminf_{n \to \infty} R_n(s \mid t)^{1/n} \geq \sup_{\varphi} \frac{\varphi(s)}{\varphi(t)}.$$

*Proof.* If $R_n(s \mid t) = \infty$ there is nothing to prove. Otherwise, by tensor-power scaling and the cover axiom, $\varphi(s)^n = \varphi(s^{\otimes n}) \leq \sum_j \varphi(u_j) \leq N \varphi(t^{\otimes n}) = N \varphi(t)^n$ with $N = R_n(s \mid t)$, since $u_j \preceq t^{\otimes n}$ gives $\varphi(u_j) \leq \varphi(t^{\otimes n})$. Taking $n$th roots gives the second claim, since $\varphi(t) \geq 1$ for $t \neq 0$. The statement for the limit inferior follows by taking the supremum over $\varphi$. $\square$

This is the elementary direction of spectral duality. The converse requires the full duality theorem, which is Inherited Result 8.7, and it holds there in the form with several target tensors and an upper functional.

**Proposition 18.3** (From Poincaré to a Spectral Inequality)**.** *Assume the edge form of Corollary 17.6. Let $\sigma \geq 0$ be a load on pairs of $U_h$ and let $\|\sigma\| = \big(\frac{b(b-1)}{2} \sum_e \sigma_e^2\big)^{1/2}$. For every price on $U_h$ there is a load $y \geq 0$ on the edges with $\|y\|_* = \big(\frac{b}{2} \sum_{edges} y_e^2\big)^{1/2} = \sqrt{C_0'}\, \|\sigma\|$ such that*

$$\sum_e \sigma_e |e| \leq \sum_{edges\ e} y_e |e|.$$

*Proof.* If $|e| = 0$ on all edges, then by Corollary 17.6 it vanishes on all pairs and both sides are zero, and $y = 0$ would give $\|y\|_* = 0$, which is the claim when $\sigma = 0$, while for $\sigma \neq 0$ any $y$ of the required norm serves. Otherwise, by the Cauchy–Schwarz inequality and Corollary 17.6,

$$\sum_e \sigma_e |e| \leq \|\sigma\| \Big( \tfrac{2}{b(b-1)} \sum_e |e|^2 \Big)^{1/2} \leq \|\sigma\| \Big( C_0' \tfrac{2}{b} \sum_{\text{edges}} |e|^2 \Big)^{1/2}.$$

Let $y$ be proportional to $(|e|)$ on the edges with $\|y\|_* = \sqrt{C_0'}\|\sigma\|$. Equality holds in the Cauchy–Schwarz inequality for $y$ and $(|e|)$, so $\sum_{\text{edges}} y_e|e| = \|y\|_*(\frac{2}{b}\sum_{\text{edges}}|e|^2)^{1/2}$, which is the right-hand bound. □

**Definition 18.4** (Law)**.** Let $A$ be a commutative ring. A law on $U_h$ over $A$ consists of a finite family of pairs of distinct elements of $U_h$, the slots, counted by a load $\sigma$ on pairs; a finite nonempty set $\mathcal{I}$ of indices with an injective code into strings of $\lambda \geq 1$ bits; for every $i \in \mathcal{I}$ a load $\text{len}_i$ on the edges, called the channels; and for every $i \in \mathcal{I}$ and $g \in U_h$ a function with values in $A$, the node tensor of $(i, g)$, of the bits at $g$ of the slots containing $g$ and of the bits of the channels containing $g$. These are required to satisfy, for all values of the bits at the ends of the slots, that the product over slots of the indicators that the $\sigma_e$ bits agree at both ends equals the sum over $i \in \mathcal{I}$ and over the contents of all channels of the product over $g \in U_h$ of the node tensors.

**Inherited Result 18.5** (Existence of Laws)**.** *There are a finitely generated subring $A$ of the ground field, a positive integer $a_1$, and finitely many laws on $U_h$ over $A$ such that, with the norm $\|n\| = (\frac{b(b-1)}{2}\sum_e n_e^2)^{1/2}$ on loads on pairs, (a) for every load $n$ on pairs and every $t \in \mathbb{N}$ with $\|n\| \leq a_1 t$ some law has $n \leq t\sigma$ entrywise, and (b) for every law and every index $i$, $\|\lambda + \text{len}_i\|_* \leq \sqrt{C_1 a_1}$. This is Theorem G.2 of the manuscript. Its proof applies Inherited Result 8.7 to the inequality of Proposition 18.3, after rounding the loads to integers, and reads the resulting cover as a law.*

**Inherited Result 18.6** (Tower of Groups)**.** *For every $n \geq 1$ with $M = |G_n|$, $G_n = \text{SL}_h(\mathbb{F}_2[s]/(s^{n+1}))$, there are $G_n$-sets $V_0 = G_n, V_1, \ldots, V_J$ with equivariant $b$-to-one projections such that, for every type $i \in \mathbb{Z}/h$, the generators other than the $i$th generate a copy of $U_h$ acting freely on every $V_j$, the sets $V_J$ have bounded size, $|G_{n+1}| = 2^{h^2-1}|G_n|$, and every element of $G_n$ is a word of length $O((n+1)^7)$ in the generators. This is Proposition H.1.*

**Inherited Result 18.7** (Decomposition into Tree Networks)**.** *There is a constant $C$ such that, for a network over a commutative $A$-algebra with at most $M$ nodes and $N$ packets of $\ell$ bits at each node, the value is a sum over sequences of good outcomes of the values of networks with the following properties, with $\Lambda = (\lg M)^C$ and $\gamma = 1/\log b$. The nodes form a rooted tree of depth at most $\Lambda$, each node but the root having at most $b$ children. At most $M^C$ named dead packets may join any two nodes, and every other packet joins a node and its parent, with at most $\Lambda M^{1-\gamma}(N + M^2)$ of them at one node and its children and at most $\Lambda M(N + M^2)$ in all. The structure of each network is computed from the outcomes by a straight-line program with at most $(MN)^C$ steps that does not depend on the contents of the packets or on the node tensors. This is Theorem H.2, whose parts (c) to (e) are used only in the cutting of computations.*

**Proof Obligation 18.8** (Machine-Verified Components)**.** *The exactness of the straight-line program in Inherited Result 18.7, and the details of the packet count, are established in the source only by a description of the program and an accounting of its size, with the proof*

*of correctness given in the Lean development. An independent reconstruction must either reproduce the program or audit the formal proof.*

# Chapter 19

# Computations as Weighted Networks

## 19.1 Weighted Nondeterministic Programs

The contracting case treats computations as the objects whose values are to be decomposed. The model is a nondeterministic program on binary trees, in which a tree is either nil or a pair of trees. A program is built from a small table of constructs: fixed trees, the argument, projections, pairing, a conditional, a call of a program coded by the output of another, a nondeterministic guess of a bit, failure, multiplication by a named generator of the coefficient ring, and a construct that appends a name to a tally. A run carries a weight, which begins at one and is multiplied by the generators that the run names, and a tally, which lists the names it has used. A run is accepting if it does not fail, and the sum of an instance is the sum of the weights of its accepting runs [30]. Any machine model with a self-interpreter of constant overhead would serve, and polylogarithmic factors are absorbed into powers of the scale parameter that appears in the later chapters.

The model is chosen so that a computation is a weighted sum over its accepting runs. This places it in the same family as the permanent and the network polynomial, which are weighted sums over combinatorial structures, and it is what allows the algebraic machinery of tensor networks to be brought to bear on computation itself.

## 19.2 Plain Instances

An instance is a pair of a program and an argument. It is plain, for a time bound $\tau$, if all of its runs halt and every accepting run has an empty tally and costs at most $\tau$. The tally is used only in the construction of accelerators, where a program that simulates a circuit over the real numbers cannot contain the constants of the circuit and names them in the tally instead. Plain instances name no constants, and they are the instances to which the decomposition into tree networks applies without modification. The distinction between plain and non-plain instances governs the structure of the final argument, because

the program that is to be speeded up contains the accelerator, which does name constants, and a lemma that collects the constants in front of the sum over runs is needed to reduce such programs to plain ones.

## 19.3 Universal Computation

A universal automaton evaluates the sum of every plain instance with a number of steps that is nearly quadratic in the size of the instance plus the time bound. The automaton is a tape automaton, a weighted finite automaton that also writes on tapes. It is obtained from a pointer machine that evaluates the program on its argument, with a stack of frames kept in the heap, by guessing what every load from memory returns and checking all the guesses at the end, in the classical manner of simulating a random-access machine by a machine with tapes. On a plain instance the machine never uses a name, since an accepting run of a plain instance has an empty tally, and so the value of the automaton equals the sum. The statement of the proposition is simple and its proof is a long construction whose exactness is, according to the source, proved only in the Lean development.

## 19.4 Computational Histories as Tensor Networks

The runs of a tape automaton form a tensor network once the computation is cut into time blocks and the input is cut into parts. The nodes of the network are the time blocks together with the parts of the input, and the wires carry the states and the tape contents that are passed between them. Data that say, for every tape and every node, whether the node writes anything and which nodes read what it writes, are chosen so that the history is split consistently. The value of the automaton in the given number of steps is the sum of the values of networks on these nodes, one network for every valid choice of the data. This representation places the computation in the setting of the previous chapter, where any tensor network is a sum of tree networks of small depth, and the tree shows where to cut the computation [30].

## 19.5 Cutting Computations into Pieces

A cut replaces a computation by a sum of products of smaller computations. The cutting program takes an instance, runs the decomposition of the previous chapter on the network of its history, and lists for every accepting run an instance for each node of the resulting tree; the original sum is then the sum over accepting runs of the weight of the run times the product of the sums of the listed pieces. A program cuts a pair of parameters bounding the time and the size into smaller parameters with a bounded number of pieces in a bounded number of steps, and it must halt on every instance of the given size, plain or not, a requirement that is used when the cutting program is applied to programs that contain accelerators. The theorem on one cut states that a cut by a factor $k$ divides the time and the size by $k$ up to a

polylogarithmic factor, with at most $k$ times that factor pieces. Iterating such cuts, with the polylogarithmic factors absorbed into the parameter $v$, reduces a computation of any length up to a power of $v^H$ to pieces of fixed time $v^H$ and size $v^{H-32}$. This is the mechanism by which a device that speeds up computations at one scale is made to speed up computations at every larger scale.

## 19.6 Formal Statements

**Definition 19.1** (Weighted Program and Sum)**.** Let $A$ be a commutative ring. A program is an expression built from the constructs of Section 19.1, and a run carries a weight in $A$, equal to 1 at the start and multiplied by the generators of $A$ that the run names, and a tally, a list of names. A run is accepting if it does not fail. The sum of an instance $z$ is $\mathrm{sum}(z) = \sum_{\rho \in \mathrm{Acc}(z)} w(\rho)$, where $w(\rho)$ is the weight of the accepting run $\rho$.

**Definition 19.2** (Plain Instance)**.** An instance $z$ is $\tau$-plain if all its runs halt and every accepting run has an empty tally and costs at most $\tau$ steps.

**Inherited Result 19.3** (Universal Tape Automaton)**.** *There are a tape automaton and a constant $C_{\mathrm{univ}}$ such that the value of the automaton on the word of an instance $z$, in $t$ steps, is* $\mathrm{sum}(z)$ *whenever $z$ is $\tau$-plain and $t \geq C_{\mathrm{univ}}\big((|z|+\tau)\lg(|z|+\tau)\big)^2$. This is Proposition I.1 of the manuscript, whose proof describes the automaton and counts its steps and whose exactness is established only in the Lean development.*

**Inherited Result 19.4** (Runs as Networks)**.** *Let an input word be cut into $K$ consecutive parts and the computation into $K$ consecutive time blocks. The value of the automaton in $KT$ steps is the sum, over valid data of reading and writing, of the values of tensor networks on the $2K$ nodes consisting of the blocks and the parts. This is Lemma I.2.*

**Definition 19.5** (Cutting Program)**.** A program $G$ cuts $(\tau, s)$ into $(\tau', s')$ with $P$ pieces in $t$ steps if, for every instance $z$ with $|z| \leq s$, all runs of $G$ on $z$ halt, the accepting ones take at most $t$ steps, have empty tallies and output lists of at most $P$ instances of size at most $s'$; and if $z$ is $\tau$-plain, then all these instances are $\tau'$-plain and

$$\mathrm{sum}(z) = \sum_{\text{accepting runs}} \text{weight} \cdot \prod_{y \text{ in the list}} \mathrm{sum}(y).$$

**Inherited Result 19.6** (One Cut)**.** *There are constants $a_0, C, v_0$ such that, with $\Lambda = (\lg v)^C$, for all $v \geq v_0$, all $\tau \geq v^{a_0}$ with $4\lg\tau \leq (\lg v)^2$, all $s \leq \tau$ and all $2 \leq k \leq v^{64}$, some program of size at most $v^{a_0}$ cuts $(\tau, s)$ into $\big(\tau\Lambda/k, (s/k + \tau/v^{256})\Lambda\big)$ with at most $k\Lambda$ pieces in $(s + \tau k/v^{256})\Lambda$ steps. This is Theorem J.3, which rests on Proposition J.2 on the pieces of a computation and on the machine constructions of Lean.*

**Inherited Result 19.7** (Cuts at All Scales)**.** *There are a multiple $H \geq 272$ of 16 and constants $C_{\mathrm{cut}}, v_0$ such that, whenever $v \geq v_0$ and $\alpha$ satisfy $(\lg v)^{C_{\mathrm{cut}}\alpha} \leq v$, for all $\tau$ and*

*$s$ with $v^H \le \tau \le v^{H\alpha}$ and $sv^{48} \le \tau$, some program of size at most $v^H$ cuts $(\tau, s)$ into $(v^H, v^{H-32})$ with at most $\tau v^2/v^H$ pieces in at most $\tau/v^{24}$ steps. This is Theorem J.4, obtained by composing $O(\alpha)$ cuts of Inherited Result 19.6. Its exponents are inherited from the cutting construction and have not been rederived here.*

**Proof Obligation 19.8** (Printed Machine Constructions)**.** *Proposition I.1, parts of Theorem H.2, Theorem J.3 and Proposition N.4 of the source describe machines without printing them, and their exactness is proved only in the Lean development. These are mandatory verification obligations for an independently maintained treatment.*

# Chapter 20

# Accelerators and the Consequences of Small Circuits

## 20.1 The Universal Computation Polynomial

The step that connects arithmetic circuits for the permanent to computation is to encode a computation as a polynomial. A word of fewer than $n$ symbols is encoded by an indicator assignment of variables $y_{i,\sigma}$ indexed by place and symbol, where an end marker counts as a symbol, and the polynomial $\Phi$ sums over all words $w$ the value $U_t(w)$ of the universal automaton on $w$ in $t$ steps times a monomial $y^w$ that picks out the symbol at each place and the end marker after the last. Because the monomial is the product of exactly the variables that are equal to one at the indicator assignment of $w$, and no indicator assignment contains the variables of another, the polynomial evaluates at the indicator assignment of a word to the value of the automaton on that word. The polynomial is a table of the computation, written as a sum over inputs of the output times an indicator of the input, and its circuit complexity is the amount of information needed to describe the computation as a polynomial.

## 20.2 Reduction to the Permanent

The polynomial $\Phi$ is a clause sum, in the terminology of Chapter 14. Its bits are the indicator assignment of the word, for every step a one-hot code of the state and of the letter, for every step and every side of each tape a one-hot counter of the symbols written so far, and for every tape and place a one-hot code of a symbol. Clauses of weight zero enforce one-hotness, the first and last states, and the single end mark. A clause sum is a permanent of moderate size with entries affine in one variable each, and consequently a circuit for the permanent of a sufficiently large matrix yields a circuit for $\Phi$ of comparable size. The lower bound is obtained by contradiction: if the permanent had a small circuit, then there would be a small description, in the form of a circuit, of the entire universal computation up to a stated length, and this description is what will be shown to be too powerful to exist.

## 20.3 Low-Degree Evaluation

A circuit for $\Phi$ can be used to evaluate $\Phi$ at an input, but evaluating the whole circuit would take time proportional to its size, which is much longer than the time of the computation being simulated. The remedy is that at an indicator assignment of a word only the part of $\Phi$ of low degree matters, since the word has few symbols. The evaluation of the part of degree at most $d$ of a polynomial with a circuit of $L$ gates can be done by a nondeterministic program whose accepting runs take about $d$ plus the size of the instance steps, up to polylogarithmic factors in $L$, by guessing the parse tree of the circuit restricted to low degree. The program has size of order $\log L + \log n + d$ and carries polynomially many constants, the constants of the circuit, which it names but does not multiply.

## 20.4 The Hypothetical Accelerator

An accelerator at a scale $v$ is a program that evaluates the sum of every short plain instance in time $v^{H-16}$, where short means of size at most $v^{H-32}$, whereas the universal automaton would need time $v^H$ and more. It is allowed to name constants of the ground field, and every name that it uses must stand for a power $\xi^{2^i}$ of a constant with $i$ less than an exponent bound. The constants sit at the leaves of a binary trie, and names are paths in the trie, so that a program can multiply by $\xi^e$ by naming $\xi^{2^i}$ for the binary digits $i$ of $e$ that are one. A small circuit for the permanent would give an accelerator of size a power of $v$ with exponent proportional to the logarithm of the circuit size. The hypothesis of a small circuit is thus converted into the existence of a computational device that is faster than the computation it simulates, and the contradiction will come from applying such a device to itself.

## 20.5 The Problem of Named Constants

The cutting theorem of Chapter 19 applies only to plain instances, whereas the program that is to be sped up at a larger scale contains the accelerator and therefore names constants. A lemma on postponing constants resolves this. A program whose accepting runs halt within a given number of steps and name constants among the leaves of a trie has a value that is a combination $\sum_e \lambda^e \operatorname{sum}(z_e)$ of sums of plain instances $z_e$, where $e$ runs over vectors of exponents, $\lambda^e$ is the corresponding product of powers of constants, and the instance $z_e$ accepts exactly the runs of the original program in which the tally uses each name the prescribed number of times. The instance $z_e$ is plain within a time that is the original time multiplied by a polylogarithmic factor. The constants have been moved in front of the sum, and what remains inside is a sum of plain instances to which the cutting theorem applies.

## 20.6 The Accumulation of Transformation Costs

The transfer of an accelerator from the scale $v$ to the larger scale $v' = v^{\alpha}$ is carried out in rounds, each of which lowers the exponent of $v$ in the running time by eight. In every round the program guesses a vector of exponents for its constants, forms the plain instances, cuts them to the scale $v$ by the cutting theorem, and runs the accelerator on every piece. After a number of rounds proportional to $\alpha$ the running time is at most $v'^{H-17}$, and the price is a growth in the program, which acquires the cutting program and the previous program at each round, and a growth in the exponent bound for the constants, which increases by the length of the names at each round. Both increases are controlled by powers of $\alpha \lg v$, and the hypothesis that a fixed power of $\alpha \lg v$ is at most $v$ is what keeps these costs below the scale. The constants of the original accelerator are preserved through the transfer, and it is this preservation that allows the diagonal argument of the next chapter to be carried out in a fixed finite field.

## 20.7 Formal Statements

**Definition 20.1** (Word Encoding)**.** For a word $w = w_1 \cdots w_r$ over a finite alphabet, with $r < n$, let $y^w = \left(\prod_{i=1}^{r} y_{i,w_i}\right) y_{r+1,\mathrm{end}}$, where the variables are $y_{i,\sigma}$ for places $1 \le i \le n$ and symbols $\sigma$, an end marker counting as a symbol. The indicator assignment $e(w')$ of a word $w'$ of fewer than $n$ symbols sets $y_{i,w'_i} = 1$ for $i \le |w'|$ and $y_{|w'|+1,\mathrm{end}} = 1$, and all other variables to 0.

**Definition 20.2** (Universal Computation Polynomial)**.** $\Phi_{n,t}(y) = \sum_{|w|<n} U_t(w)\, y^w$, where $U_t(w)$ is the value of the universal automaton on the word $w$ in $t$ steps.

**Proposition 20.3** (Evaluation Identity)**.** *For every word $w'$ with $|w'| < n$, $\Phi_{n,t}(e(w')) = U_t(w')$.*

*Proof.* The monomial $y^w$ equals 1 at $e(w')$ exactly when all its variables are 1, that is, when $w_i = w'_i$ for all $i \le |w|$ and the variable $y_{|w|+1,\mathrm{end}}$ is 1 at $e(w')$. The latter holds exactly when $|w| + 1 = |w'| + 1$. So $y^w(e(w')) = 1$ if $w = w'$ and 0 otherwise. In particular a proper prefix $w$ of $w'$ does not contribute, because $y_{|w|+1,\mathrm{end}} = 0$ at $e(w')$, the place $|w| + 1$ holding a symbol of $w'$. Hence only $w = w'$ contributes, with the value $U_t(w')$. □

**Proposition 20.4** (Degree Bound)**.** $\deg \Phi_{n,t} \le n$.

*Proof.* The monomial $y^w$ has degree $|w| + 1 \le n$ because $|w| < n$. □

**Inherited Result 20.5** (Completeness Reduction)**.** *There are constants $B_0, C, v_0$ such that for all integers $a \ge 0$ and $v \ge v_0$ with $(\lg v)^{Ca} \le v$, if $R$ is a field that is an $A$-algebra with $2 \ne 0$ in $R$ and $\mathrm{PERM}_m$ has a circuit over $R$ with at most $v^a$ gates for some $m \ge v^{B_0}$, then $R$ has an accelerator of size $v^{C(a+1)}$ at scale $v$. This is Lemma K.3, which combines the clause-sum realization of $\Phi_{n,t}$ as a permanent (Lemma E.3 and Corollary E.2) with the evaluation of the low-degree part of a circuit (Lemma K.2).*

**Inherited Result 20.6** (Low-Degree Evaluation)**.** *Let $R$ be a commutative $A$-algebra, $d \geq 0$, and let a polynomial over $R$ in $n$ places of variables have a circuit with $L$ gates. There is a program of size $O(\log L + \log n + d)$ with $O((L+n)^3)$ constants that evaluates the part of degree at most $d$ at indicator assignments in $d$ plus the size of the instance steps up to polylogarithmic factors in the accepting runs. This is Lemma K.2.*

**Definition 20.7** (Accelerator)**.** Let $R$ be a commutative $A$-algebra. An accelerator at scale $v$ with constants $\xi_1, \xi_2, \ldots \in R$ and exponent bound $B$ is a program $D$ such that (i) on every instance $z$ with $|z| \leq v^{H-32}$ all runs of $D$ halt, the accepting ones within $v^{H-16}$ steps, and every name in the tally of an accepting run stands for a power $\xi_l^{2^i}$ with $i < B$; and (ii) if moreover $z$ is $v^H$-plain, the value of $D$ on $z$, the sum over accepting runs of the weight times the powers named in the tally, each as often as it is named, is $\mathrm{sum}(z)$. The constants sit at the leaves of a binary trie with $Q$ nodes, the name of $\xi_l^{2^i}$ being the path to the leaf of $\xi_l$ followed by $1^i0$. The accelerator has size $s$ if $Q$ and the size of $D$ are at most $s$ and, with $\beta = \lg \max(v^H, s)$, the height of the trie is at most $\beta^2$ and $B \leq \beta^3$.

**Inherited Result 20.8** (Postponing the Constants)**.** *Let $p$ be a program and an argument on which all runs of $p$ halt, the accepting ones within $T < 2^J$ steps and with names among the leaves of a binary trie with $Q'$ nodes and height $\eta$, the leaves standing for powers $\lambda_1, \lambda_2, \ldots$. Then the value of $p$ on the argument is $\sum_e \lambda^e \,\mathrm{sum}(z_e)$ with $\lambda^e = \prod_j \lambda_j^{e_j}$, over the vectors $e$ with entries in $\{0, \ldots, 2^J - 1\}$, and the instance $z_e$ is plain within $O(T(J+\eta+1)+Q'(J+1))$ steps, of size that of $p$ and the argument plus $O(Q'(J+1))$. This is Lemma K.4.*

**Inherited Result 20.9** (Transfer to the Next Scale)**.** *Let $v$ be large, $\alpha \geq 5$ an integer with $(\lg v)^{C\alpha} \leq v$, and $D$ a program with the two properties of an accelerator at scale $v$ whose trie has $Q \leq v^{h_0\alpha}$ nodes and whose size is at most $v^{h_0\alpha}$, with $h_0 = H/16$, the exponent bound and trie height at most a fixed power of $\alpha \lg v$. Then there is a program with the two properties of an accelerator at scale $v' = v^\alpha$, with the same constants, exponent bound $B + 51h_0\alpha^2 \lg v$, size at most $v'^{4h_0}$, and accepting runs of at most $v'^{H-17}$ steps. This is Proposition K.5.*

**Proposition 20.10** (Accumulation of the Exponent Bound)**.** *In the setting of Inherited Result 20.9, with $J = (H\alpha + 1)\lg v$ and at most $3\alpha$ rounds, each adding $J$ to the exponent bound, the total increase $3\alpha J$ is at most $51h_0\alpha^2 \lg v$.*

*Proof.* With $H = 16h_0$ one has $3\alpha J = 3\alpha(16h_0\alpha + 1)\lg v = 48h_0\alpha^2 \lg v + 3\alpha \lg v$, and $3\alpha \lg v \leq 3h_0\alpha^2 \lg v$ because $h_0 \geq 1$ and $\alpha \geq 1$. □

# Chapter 21

# Diagonalization and the Impossibility of Universal Acceleration

## 21.1 Self-Application

The contradiction is obtained by applying an accelerator to a program that is built from the accelerator itself. Given an accelerator $D$ at a large scale, one writes a program $P$ that, on an argument $x$, guesses one bit. For one value of the bit it accepts with weight one. For the other value it runs $D$ on the instance consisting of the program coded by $x$ and the argument $x$, after multiplying by an appropriate constant power. At the argument $x = P$, the program is applied to its own code. Two facts are then simultaneously true of the value $V$ of $P$ on $P$. By the definition of $P$, it is the sum of the contribution 1 of the first branch and the contribution of the second branch, which is the value that $D$ computes for the instance $(P, P)$. By the correctness of $D$ on short plain instances, that value is the value of $P$ on $P$, which is $V$ itself. The result is the equation $V = 1 + V$.

## 21.2 Reduction to a Finite Field

The construction requires that the ring in which the values live have finitely many constants of a controlled kind, and this is achieved by passing to a finite field. The ring generated over $A$ by the constants of the accelerator is finitely generated, and modulo a maximal ideal a finitely generated nonzero commutative ring is a field that is finitely generated as a ring, hence finite. Nothing is known about the size of the resulting field $E$, and the diagonal argument is arranged so that it does not need to be. The accelerator remains an accelerator over $E$, because the identities that define it are polynomial identities in the constants. The property of a finite field that is essential for the argument is that the iterated squarings of its elements repeat: for each $j$ the map $c \mapsto c^{2^j}$ coincides with an earlier one of the sequence,

and there are only finitely many distinct maps. This allows every power of the constants that the program must name to be named with a bounded exponent, which is what keeps the exponent bound of the diagonal program at the scale allowed by the definition of an accelerator.

## 21.3 The Contradictory Computation

The scale at which the diagonal program is built is chosen large relative to the size of the field and the number of constants, by the transfer of accelerators to larger scales applied repeatedly with the same constants. At such a scale the accelerator has size $v^{4h_0}$ and exponent bound $O((\lg v)^3)$, the diagonal program $P$ has size about $v^{H/4}$, so that every instance that the accelerator is asked to evaluate in the diagonal computation is short, and the accelerator therefore halts on all of them within $v^{H-16}$ steps, accepting. The program $P$ then halts on all runs, the accepting ones within $v^{H-15}$ steps, and by the postponement of constants its value is a combination of sums of plain instances, to each of which the correctness of the accelerator applies. The identity $V = 1 + V$ follows, and it is impossible in a field. The elementary contradiction is not the difficult part. The difficulty lies in proving that the self-applied program satisfies every condition on size, halting, and exponent bound that the accelerator's correctness requires, and that is the content of the transfer to larger scales together with the accounting of the postponement.

## 21.4 The Contracting-Case Lower Bound

Combining the reduction of Chapter 20 with the nonexistence of accelerators gives the lower bound for the permanent in the contracting case. A circuit with at most $2^{c(\log m)^2/\log\log m}$ gates, for small enough $c$, produces an accelerator at the scale $v = \lfloor m^{1/B_0} \rfloor$ of size $v^{C(a+1)}$, where $v^a$ bounds the number of gates, and the condition that $(\lg v)^{C'(a+1)} \leq v$ holds because $a$ is of order $c \log m / \log\log m$, so that the left side is $2^{O(c \log m)}$ and the right side is $2^{\Omega(\log m)}$. The accelerator contradicts the nonexistence theorem. The bound that results, $2^{c(\log m)^2/\log\log m}$, exceeds the flat-case bound $2^{c(\log m)^{4/3}}$ for large $m$, so that in the final assembly the constant of the flat case governs. The constant in the contracting case depends on parameters, namely the constants of the cutting and of the accelerator, that are not known explicitly, and this is the reason that no value of $m_0$ is available for that case.

## 21.5 Why the Two Cases Are Conceptually Different

The flat case and the contracting case reach the same conclusion by different kinds of reasoning, and the difference should be stated openly. The flat case is a direct representational obstruction. A small circuit induces a decomposition of the coefficient tensor along balanced partitions, and a geometry in which the ends of the wires are placed at points of a Euclidean space shows that any such decomposition costs more than the network allows. The quantity

that is bounded is the size of a representation, and the bound is obtained by comparing the lengths of two ways of writing the same tensor. The contracting case proves no bound on any single representation. It shows that a hypothetical economy of representation, in the form of a small circuit for the permanent, would yield a device that evaluates all computations faster than they run, and that such a device would, applied to itself, assert that a value equals one more than itself. The obstruction is by computational self-reference: the lower bound is the statement that the economy would be self-contradictory, and the geometry enters only through the rerouting that makes the speed-up transfer across scales. One case obstructs by a measurement and the other by a diagonal argument, and the quantitative content is correspondingly different: an explicit constant in the flat case, a constant tied to unspecified machine parameters in the contracting case. The theorem uses both, because the dichotomy of Chapter 13 provides no information about which case occurs, and this is the sense in which its logical completeness exceeds its explanatory completeness.

## 21.6 Formal Statements

**Definition 21.1** (Hypothetical Accelerator)**.** Let $\mathcal{A}$ be a program that is an accelerator at scale $v$ over a nonzero commutative $A$-algebra $R$ in the sense of Definition 20.7. Its correctness requirement is $\mathrm{Val}(\mathcal{A}, z) = \mathrm{sum}(z)$ for every short $v^H$-plain instance $z$.

**Inherited Result 21.2** (Finite-Field Specialization)**.** *If a nonzero commutative $A$-algebra has an accelerator of a given size at scale $v$, then so does some finite field $E$ that is an $A$-algebra, with the same program, constants reduced modulo a maximal ideal, and the same size. This uses that a nonzero finitely generated commutative ring has a maximal ideal with finite residue field, and that the defining properties of an accelerator are polynomial identities in the constants. It is used in the proof of Theorem K.6.*

**Proposition 21.3** (Finite Repetition of Squarings)**.** *Let $E$ be a finite field and $m_E = |E|^{|E|}$. For every $j \geq 0$ there is $r(j) < m_E$ with $c^{2^j} = c^{2^{r(j)}}$ for all $c \in E$.*

*Proof.* Let $f_j : E \to E$ be the map $c \mapsto c^{2^j}$. There are at most $|E|^{|E|} = m_E$ maps from $E$ to $E$, so the sequence $(f_j)_{j \geq 0}$ takes at most $m_E$ distinct values. Let $r(j)$ be the least index $r$ with $f_r = f_j$. The least indices of distinct values are distinct, so they number at most $m_E$ and each is less than $m_E$. □

**Inherited Result 21.4** (Self-Application Construction)**.** *Let $E$ be a finite field, $D$ an accelerator over $E$ at a scale $v$ large compared with $|E|$ and the number of constants, obtained by transfer to larger scales with the same constants, with exponent bound $B$ and $2^J > v^{H-15}$. Let $B' = \max(B, m_E)$ and let $\lambda$ be the list of the powers $\xi_l^{2^i}$ with $i < B'$. There is a program $P$ of size at most $v^{H/4}$ plus lower-order terms that, on an argument $x$, guesses a bit, accepts with weight 1 for one value and for the other guesses an exponent vector $e$ for $\lambda$, multiplies by $\lambda^e$ by naming, and runs $D$ on the instance $z_e$ that Inherited Result 20.8 provides for the program $x$ on the argument $x$. All runs of $P$ on $P$ halt, the accepting ones within $v^{H-15}$*

*steps, and the value of $P$ on $P$ is $V = 1 + \sum_e \lambda^e \operatorname{sum}(z_e)$. This is the construction in the proof of Theorem K.6, using Proposition 21.3 to name every $\lambda^e$ with exponent bound $B'$.*

**Proposition 21.5** (Contradiction)**.** *In a nonzero commutative ring there is no element $V$ with $V = 1 + V$.*

*Proof.* Subtracting $V$ gives $0 = 1$, so that every element $r$ equals $r \cdot 1 = r \cdot 0 = 0$ and the ring is zero. □

**Inherited Result 21.6** (Nonexistence of Accelerators)**.** *There are $C$ and $v_1$ such that for all integers $a \geq 0$ and $v \geq v_1$ with $(\lg v)^{Ca} \leq v$, no nonzero commutative $A$-algebra has an accelerator of size $v^a$ at scale $v$. This is Theorem K.6. Given Inherited Results 21.4 and 20.9, the value of $P$ on $P$ is $V = \sum_e \lambda^e \operatorname{sum}(z_e)$ by postponement of constants, and equals $1 + \sum_e \lambda^e \operatorname{sum}(z_e) = 1 + V$ by the correctness of $D$ on the short plain instances $z_e$, which contradicts Proposition 21.5 in the field $E$.*

**Theorem 21.7** (Contracting-Case Lower Bound)**.** *Assume the contracting condition and Inherited Results 20.5 and 21.6. There are $c > 0$ and $m_0$ such that for $m \geq m_0$ every circuit over the ground field for $\mathrm{PERM}_m$ has more than $2^{c(\log m)^2/\log\log m}$ gates.*

*Proof.* Suppose, for a small $c > 0$ and some large $m$, that a circuit has at most $2^{c(\log m)^2/\log\log m}$ gates. Let $v = \lfloor m^{1/B_0} \rfloor$, so that $m \geq v^{B_0}$, and let $a$ be least such that the circuit has at most $v^a$ gates. Since $\log v \geq (\log m)/B_0 - 1$, $a \leq \dfrac{c(\log m)^2/\log\log m}{\log v} + 1 \leq \dfrac{2cB_0 \log m}{\log\log m} + 1$ for large $m$. Then $(\lg v)^{C'(a+1)} = 2^{C'(a+1)\log\lg v} \leq 2^{4C'cB_0 \log m + O(\log\log m)}$, and this is at most $v = 2^{\Omega(\log m)}$ once $c$ is small enough that $4C'cB_0 < \frac{1}{2B_0}$ and $m$ is large. By Inherited Result 20.5 there is an accelerator of size $v^{C(a+1)}$ at scale $v$ over the ground field, a nonzero $A$-algebra, which contradicts Inherited Result 21.6 with $C(a+1)$ in place of $a$. □

**Proposition 21.8** (Dominance of the Contracting Bound)**.** $\displaystyle\lim_{m\to\infty} \frac{(\log m)^2/\log\log m}{(\log m)^{4/3}} = \infty$. *Consequently $2^{c'(\log m)^2/\log\log m} \geq 2^{c(\log m)^{4/3}}$ for every $c, c' > 0$ and all sufficiently large $m$.*

*Proof.* With $x = \log m$ the quotient is $x^{2/3}/\log x$, which tends to infinity, since $\log x$ grows more slowly than any positive power of $x$. □

*Remark* 21.9. The unknown constants of the cutting construction and of the accelerator mean that Theorem 21.7 gives no value of $m_0$ and a constant $c$ that is not explicit, and over a field of positive characteristic it depends on the characteristic. The assembly of the absolute constant and the quantifier structure of Chapter 6 therefore rests on the flat-case rate in the dominant position and on the field-transfer lemma of the source.

# Part VIII

# Boundaries of the Theorem

# Chapter 22

# Why the Argument Does Not Resolve P Versus NP

## 22.1 The Boolean Quotient

A Boolean circuit for the Hamiltonian cycle polynomial is a different object from an arithmetic circuit for it, and the difference is exactly the kernel of the evaluation map met in Chapter 3. An arithmetic circuit over $\mathbb{F}_2$ that is correct on the inputs 0 and 1 is not required to compute $\mathrm{HC}_m$ as a polynomial. It may compute any polynomial $g = \mathrm{HC}_m + \sum_i (x_i^2 - x_i) h_i$, and the manuscript's theorem is a statement about circuits that compute $\mathrm{HC}_m$ itself. The ideal $I_B = \langle x_i^2 - x_i \rangle$ therefore measures the freedom that a Boolean circuit has and an arithmetic circuit lacks. The relevant optimization, over the whole coset, is one that the lower bound does not touch, and the purpose of this chapter is to locate precisely the steps of the proof at which the coset enters, so that the scope of the theorem is stated as a mathematical fact and not as a disclaimer.

The source manuscript marks the dependence on the circuit's internal structure at two steps of its argument. The first is the decomposition of the polynomial computed by a small circuit along balanced partitions of the blocks of variables, which uses the part of the computed polynomial that has degree one in every block. The second is the evaluation of the universal computation polynomial, which uses the part of the computed polynomial of low degree. The substitutions that carry the clause sums to the permanent or to the Hamiltonian cycle polynomial are not obstacles, because they send inputs 0 and 1 to inputs 0 and 1, and the reduction therefore survives restriction to Boolean values. The two steps that look inside the circuit are the ones at which a circuit that is correct only on the cube can differ from one that computes the polynomial.

## 22.2 Loss of Formal Distinctions

Two polynomials in the same coset of $I_B$ agree at every Boolean point, and their formal difference is invisible to evaluation. What the extraction operators of the proof retain,

however, are components of the polynomial: the part of degree one in every block, and the part of bounded degree. These components are not constant on cosets. The standard example is $(x_1^2 - x_1)x_2 \cdots x_k$, which vanishes at every point of the cube, and whose part of degree one in each of the blocks $\{x_1\}, \ldots, \{x_k\}$ is $-x_1 x_2 \cdots x_k$, a nonzero monomial. A stronger statement holds and is worth recording: for the extraction operators used in the proof, the image of the ideal is the whole space of possible outputs, so that nothing about the extracted component can be inferred from the coset. The component of a Boolean-equivalent polynomial can be any polynomial that is linear in each block, and its top-degree component can be any homogeneous polynomial of the relevant degree. A decomposition lemma that is applied to the extracted component of a circuit's output therefore tells nothing about the component of the target polynomial, once the circuit is only known to be correct modulo $I_B$.

## 22.3 Failure of the Balanced Decomposition Argument

The balanced-products proposition of Chapter 15 is valid for every circuit, and it applies without change to a circuit that computes $g$ with $E(g) = E(\mathrm{HC}_m)$. What it yields in that case is a decomposition of the part of $g$ of degree one in every block into a sum of balanced products, and this part is the coefficient tensor of the target polynomial plus the part extracted from an element of $I_B$. The second summand is unconstrained by Proposition 22.6 below. The spectral argument of Chapters 15 and 16 bounds the size of a decomposition of the tensor $T_{k,\ell}$, and with a Boolean circuit the decomposition obtained is of a different tensor, of which nothing is known. The step that fails is the identification of the decomposed tensor with $T_{k,\ell}$, and no change in the estimates repairs it, because the failure is of an identity and not of an inequality.

## 22.4 Failure of the Low-Degree Evaluation Argument

The same occurs at the evaluation step of the contracting case. The identity $\Phi_{n,t}(e(w')) = U_t(w')$ of Chapter 20 evaluates the computation polynomial at the points that encode words, and the program extracted from a circuit computes the part of its output of degree at most $d$, which for an exact circuit is $\Phi$ itself. For a circuit that agrees with $\Phi$ only on Boolean points, the part of degree at most $d$ is again $\Phi$ plus the low-degree part of an element of $I_B$, and the latter is an arbitrary homogeneous polynomial in the top degree. The encoding points $e(w')$ are Boolean points, at which every element of $I_B$ vanishes, so that the circuit's values at the encoded words are correct. The program that the circuit produces nevertheless computes the low-degree part, and the low-degree part is the wrong polynomial. The evaluation identity is stated for $\Phi$, and its proof does not extend to $\Phi + \varepsilon'$ for an error $\varepsilon'$ that vanishes at the encoding points but not elsewhere in the support of the sum that defines the program's value.

## 22.5 The General Principle of Obstruction Under Quotienting

The two failures are instances of one principle. An obstruction proved for a cost on $X$ is an obstruction for the quotient cost on $Y = X/\sim$ only if there is a cost-controlled lift, in the sense of Chapter 3. Here the quotient is $\mathbb{F}[x]/I_B \cong \mathbb{F}^{\{0,1\}^n}$, the multilinear reduction is a canonical lift, and the question is whether the cost of the lift is controlled by the quotient cost. The monograph does not claim an answer, and the theorem gives none. What can be said is that the quotient cost can be strictly smaller than the cost of the representative, as in the example $x_1^2$ and $x_1$, and that the proof of the theorem has no mechanism to bound the quotient cost from below. This is not a defect specific to this proof. It records that the separation of VP from VNP is a statement about formal polynomials, and that statements about Boolean circuits are statements about cosets, so that an implication from the first to the second would require a new argument that the monograph does not supply. The relation between the algebraic and the Boolean hypotheses is studied in the literature only through conditional implications under additional assumptions [11], and none of them is used here.

## 22.6 Formal Statements

**Definition 22.1** (Boolean Ideal)**.** Let $\mathbb{F}$ be a field and $I_B = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle \subseteq \mathbb{F}[x_1, \ldots, x_n]$.

**Proposition 22.2** (Boolean Equivalence)**.** *For polynomials $f, g$, $E(f) = E(g)$ if and only if $f - g \in I_B$.*

*Proof.* This is Proposition 3.5, since $E$ is linear and $\ker E = I_B$. □

**Definition 22.3** (Extraction Operators)**.** Fix a partition of the variables into $k \geq 1$ nonempty blocks. The operator $M$ sends a polynomial to the sum of those of its terms that have degree exactly one in every block. For $d \geq 1$ the operator $T_{\leq d}$ sends a polynomial to the sum of its terms of total degree at most $d$. Both are linear.

**Proposition 22.4** (Failure of Descent)**.** *There are polynomials $f, g$ with $E(f) = E(g)$ and $M(f) \neq M(g)$, and with $T_{\leq d}(f) \neq T_{\leq d}(g)$.*

*Proof.* Let each variable form its own block, $k \geq 1$, and put $f = (x_1^2 - x_1)x_2 \cdots x_k$ and $g = 0$. Then $f - g \in I_B$, so $E(f) = E(g) = 0$ by Proposition 22.2. The term $x_1^2 x_2 \cdots x_k$ has degree two in the first block and is removed by $M$, while $-x_1 x_2 \cdots x_k$ has degree one in every block. Hence $M(f) = -x_1 \cdots x_k \neq 0 = M(g)$. For $d = k$ one has $T_{\leq k}(f) = -x_1 \cdots x_k \neq 0$, since the removed term has degree $k + 1$. □

**Corollary 22.5** (No Descent to the Quotient)**.** *There is no map $\bar{M}$ on $\mathbb{F}[x]/I_B$ with $\bar{M}([f]) = M(f)$ for all $f$, and likewise for $T_{\leq d}$.*

*Proof.* Such a map would give $M(f) = \bar{M}([f]) = \bar{M}([g]) = M(g)$ for the polynomials of Proposition 22.4. □

**Proposition 22.6** (Image of the Ideal under Extraction)**.** *Let $M$ be defined with respect to a partition into $k \geq 1$ nonempty blocks. Then $M(I_B)$ is the space of all polynomials that have degree one in every block and degree zero in no block, that is, the span of all monomials of degree exactly one in each block. For $d \geq 1$ the homogeneous component of degree $d$ of $T_{\leq d}(e)$, for $e \in I_B$, ranges over all homogeneous polynomials of degree $d$.*

*Proof.* By definition $M$ takes values in the span of monomials of degree one in every block, so the inclusion $M(I_B) \subseteq$ this span holds. Conversely let $m$ be such a monomial, let $x$ be a variable dividing $m$ and let $e = (x^2 - x)\, m/x = xm - m \in I_B$. The term $xm$ has degree two in the block of $x$ and is removed by $M$, so $M(e) = -m$. Since $M$ is linear and $I_B$ is a vector space, $m$ lies in $M(I_B)$. For the second claim let $m$ be a monomial of degree $d \geq 1$, let $x$ divide $m$ and take the same $e$. The term $xm$ has degree $d+1$ and is removed by $T_{\leq d}$, so that $T_{\leq d}(e) = -m$ and the claim follows by linearity. □

**Definition 22.7** (Representative Cost)**.** Let $L_{\mathbb{F}}(g)$ be the least size of a circuit over $\mathbb{F}$ computing $g$, in the sense of Chapter 3, so that variables and constants are counted as gates. Put $L_B([f]) = \inf_{g \in f + I_B} L_{\mathbb{F}}(g)$.

**Proposition 22.8** (Nontransfer)**.** *For every $f$, $L_B([f]) \leq L_{\mathbb{F}}(f)$, and the inequality can be strict. Consequently a lower bound for $L_{\mathbb{F}}(f)$ is not a lower bound for $L_B([f])$.*

*Proof.* The polynomial $f$ is a member of $f + I_B$, which gives the inequality. For $f = x_1^2$ one has $f - x_1 \in I_B$, and $L_{\mathbb{F}}(x_1) = 1$, while a list with one entry contains only a variable or a constant, so $L_{\mathbb{F}}(x_1^2) \geq 2$. Hence $L_B([x_1^2]) \leq 1 < 2 \leq L_{\mathbb{F}}(x_1^2)$. □

**Proposition 22.9** (Boolean Circuits Bound the Quotient Cost)**.** *If a Boolean circuit with $s$ gates over $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$ computes the function $E(f)$, then $L_B([f]) \leq 4s + n + 2$ over every field, where $n$ is the number of variables and the size is counted as in the definition above.*

*Proof.* By Proposition 3.6 the replacement of each gate by $xy$, $x + y - xy$ or $1 - x$ yields an arithmetic circuit computing some $g$ with $g - f \in \ker E = I_B$. The gate $xy$ becomes one entry, $x + y - xy$ becomes four entries ($xy$, $(-1)xy$, $x + (-1)xy$, and the sum with $y$) and $1 - x$ becomes two ($(-1)x$ and $1 + (-1)x$), the constants 1 and $-1$ are two further entries shared by all gates, and the $n$ variables are $n$ entries. Hence at most $4s + n + 2$ entries are used. □

*Remark* 22.10. Proposition 22.9 shows that lower bounds for $L_B$ would give lower bounds for Boolean circuits. The principal theorem is a lower bound for $L_{\mathbb{F}}$ on one representative, and Proposition 22.8 shows that the two quantities can differ. No implication from the principal theorem to a statement about Boolean circuits is derived in this monograph.

**Proof Obligation 22.11.** *Determine whether the multilinear reduction $R$, which replaces every power $x_i^j$, $j \geq 1$, by $x_i$, satisfies $L_{\mathbb{F}}(R(f)) \leq \theta(L_{\mathbb{F}}(f))$ for a function $\theta$ of polynomial growth, on the family of polynomials relevant to the theorem. By Proposition 22.2, $R(f) - f \in$*

*$I_B$ and $R$ is a section of the evaluation map on the multilinear polynomials, so this is the cost-controlled lift condition of Chapter 3. No result on it is claimed.*

# Chapter 23

# Why the Determinant Escapes the Obstruction

## 23.1 The Role of Completeness Reductions

The determinant and the permanent are given by the same sum over permutations, with and without the sign, and the determinant has circuits of polynomial size by Berkowitz's algorithm [7]. A proof that the permanent requires circuits of superpolynomial size must therefore use some property of the permanent that the determinant lacks. The source manuscript states where this occurs. The lower bounds are proved for two families of clause sums, the network polynomial in the flat case and the computation polynomial in the contracting case, and the passage to the permanent is a reduction of clause sums to the permanent, in which a small circuit for the permanent yields a small circuit for the clause sum, with the factor five of Inherited Result 14.7. Every other step concerns the clause sums. The proof of the theorem therefore establishes a lower bound for a polynomial that is not itself a permanent, and the permanent enters only as the target of a reduction.

The reduction is where completeness is used. A clause sum is a sum over bit strings of products of local weights, and the permanent is complete for such sums in the sense of Valiant: the construction of a matrix whose permanent is the clause sum uses the freedom of the permanent to encode the combinatorics of the sum in cycle covers, with entries of the form $\pm\frac{1}{2}$ that make the contributions of unwanted covers cancel. The determinant admits a construction of the same kind only for sums that it can compute, and a family of clause sums complete for VNP is not among them unless VP = VNP.

## 23.2 Structural Rather Than Superficial Differences

The two polynomials have exactly the same monomial support, as Proposition 23.1 records, so that monomial support alone cannot characterize circuit complexity, and any general theory of obstruction must be sensitive to something beyond which monomials occur. The sign of the permutations is a superficial difference between the two polynomials, in the sense

that over a field of characteristic two the sign is 1 and the two coincide. In that characteristic the principal theorem excludes the permanent for a reason that is now visible: a lower bound for the permanent in characteristic two would be a lower bound for the determinant, which has small circuits. The reduction of clause sums to the permanent uses entries $\pm\frac{1}{2}$ and fails where $2 = 0$, and for this reason the source reduces to the Hamiltonian cycle polynomial in every characteristic, by a construction that uses no division. This shows that the relevant distinction is not between the presence and absence of signs but between a polynomial that is complete for a class and one that is not. In characteristic different from two the signs do change the complexity in the way the theorem asserts, conditional on its inherited ingredients, but the mechanism is that the determinant cannot absorb the reduction, and not that signs are a barrier in themselves. The sign-removal caveat of Chapter 5 makes the same point from the side of the determinant.

## 23.3 The Location of the Proof's Failure

The location can be stated as a theorem in the flat case. Replacing the permanent in Inherited Result 14.7 by the determinant would give a reduction that the lower bound refutes. The refutation uses only the explicit lower bound of Proposition 16.8 and the polynomial size of determinantal circuits, and it is the precise sense of the manuscript's remark that its results imply that the clause sums have no reduction of comparable size to determinants. In the contracting case the same statement holds in qualitative form, since the lower bound for $\Phi$ is a bound for a clause sum, but the monograph does not state it quantitatively, because the contracting-case constants are not explicit.

The consequence for determinantal complexity is the remark of the source that, outside characteristic two, the permanent is not a determinant of an affine matrix of size $m^{c(\log m)^{1/3}}$. The exact version follows from the principal bound together with Berkowitz's theorem and is proved below. The version for approximation is Remark N.6 of the source, the one numbered statement of the manuscript that has no Lean equivalent, and it is taken up in Chapter 24.

## 23.4 Formal Statements

**Proposition 23.1** (Equal Support)**.** *For every field $\mathbb{F}$ and every $m$, the polynomials* $\det_m$ *and* $\operatorname{perm}_m$ *have the same set of monomials, namely $\prod_i x_{i,\sigma(i)}$ for $\sigma \in S_m$, each occurring with coefficient $\pm 1$.*

*Proof.* Distinct permutations give distinct monomials $\prod_i x_{i,\sigma(i)}$, because the monomial determines the column chosen in each row. Hence each monomial of $\det_m = \sum_\sigma \operatorname{sgn}(\sigma) \prod_i x_{i,\sigma(i)}$ and of $\operatorname{perm}_m = \sum_\sigma \prod_i x_{i,\sigma(i)}$ occurs with coefficient $\pm 1$, which is nonzero in every characteristic. □

**Corollary 23.2** (Support Does Not Determine Complexity)**.** *Assume Inherited Result 6.1 and* $\operatorname{char} \mathbb{F} \neq 2$*. Two families of polynomials with the same monomial support, namely* $(\det_m)$ *and* $(\operatorname{perm}_m)$*, lie on different sides of the boundary of* VP*.*

*Proof.* By Inherited Result 23.5 the determinant family is in VP, and by the principal bound the permanent family is not. □

**Definition 23.3** (Projection Closure)**.** A family $(f_n)$ is a polynomially bounded projection of a family $(h_N)$ if there are a polynomial $N(n)$ and, for every $n$, an assignment $\sigma_n$ sending each variable of $h_{N(n)}$ to a variable of $f_n$ or to a constant of $\mathbb{F}$ with $f_n = h_{N(n)} \circ \sigma_n$. The projection closure $\mathcal{R}(h)$ is the set of such families $(f_n)$.

**Proposition 23.4** (Tractable Projection Closure)**.** *If $h = (h_N) \in$ VP, then $\mathcal{R}(h) \subseteq$ VP.*

*Proof.* Let $(f_n) \in \mathcal{R}(h)$ with $N(n)$ and $\sigma_n$ as in the definition. A circuit for $h_{N(n)}$ becomes a circuit for $f_n$ when each input is replaced by the variable or constant $\sigma_n$ assigns to it, and a replaced entry is again a variable or a constant, so the list has the same length and $L_{\mathbb{F}}(f_n) \le L_{\mathbb{F}}(h_{N(n)})$. The substitution does not raise the degree, so $\deg f_n \le \deg h_{N(n)}$. Since $h \in$ VP, both $L_{\mathbb{F}}(h_N)$ and $\deg h_N$ are bounded by a polynomial in $N$, and $N(n)$ is polynomial in $n$, so $L_{\mathbb{F}}(f_n)$ and $\deg f_n$ are bounded by polynomials in $n$. □

**Inherited Result 23.5** (Determinant in VP)**.** *The family* $(\det_N)$ *lies in* VP *over every field, with a circuit of size at most $CN^{c_1}$ for absolute constants $C$ and $c_1 \ge 2$. This is Berkowitz's theorem [7], cited in the source as an input to Remark N.6.*

**Corollary 23.6** (The Permanent Is Not a Projection of the Determinant)**.** *Assume Inherited Result 6.1 and let* $\operatorname{char}\mathbb{F} \neq 2$*. Then* $\operatorname{perm} \notin \mathcal{R}(\det)$*.*

*Proof.* If $\operatorname{perm} \in \mathcal{R}(\det)$, then $\operatorname{perm} \in$ VP by Proposition 23.4 and Inherited Result 23.5. The principal bound states that $L_{\mathbb{F}}(\operatorname{perm}_m) > m^{c(\log m)^{1/3}}$ for large $m$, which is superpolynomial, so the permanent family is not in VP. □

**Proposition 23.7** (Determinantal Transfer of Lower Bounds)**.** *Let $g$ be a polynomial in $V$ variables over $\mathbb{F}$ with $L_{\mathbb{F}}(g) \ge s$. If $g = \det(A)$ for an $N \times N$ matrix $A$ whose entries are affine functions of the variables, then $N \ge \left(s/(C+5V)\right)^{1/c_1}$.*

*Proof.* Each entry $a_0 + \sum_i a_i x_i$ of $A$ is computed by a list of at most $4V+1$ entries (the constants, the variables, $V$ products and $V$ sums), so the $N^2$ entries need at most $N^2(4V+1)$ entries, which replace the variable entries of a circuit of at most $CN^{c_1}$ entries for the determinant, by Inherited Result 23.5. Hence $s \le L_{\mathbb{F}}(g) \le CN^{c_1} + N^2(4V+1) \le (C+5V)N^{c_1}$, using $c_1 \ge 2$, $N \ge 1$ and $4V+1 \le 5V$. □

**Corollary 23.8** (Exact Determinantal Complexity of the Permanent)**.** *Assume Inherited Result 6.1 with* $\operatorname{char}\mathbb{F} \neq 2$*. There are $c' > 0$ and $m_1$ such that for $m \ge m_1$, if* $\operatorname{perm}_m = \det(A)$ *for an affine $N \times N$ matrix $A$, then $N \ge m^{c'(\log m)^{1/3}}$.*

*Proof.* By Proposition 23.7 with $V = m^2$ and $s = m^{c(\log m)^{1/3}}$, $N \ge \left(m^{c(\log m)^{1/3}}/(C + 5m^2)\right)^{1/c_1}$. For large $m$, $C + 5m^2 \le m^3$ and $c(\log m)^{1/3} - 3 \ge \frac{c}{2}(\log m)^{1/3}$, so $N \ge m^{c(\log m)^{1/3}/(2c_1)}$, and one may take $c' = c/(2c_1)$. □

**Proposition 23.9** (No Determinantal Analogue of the Clause-Sum Reduction)**.** *Assume the flat condition and Proposition 16.8. There is $k_0$ such that for every $k \geq k_0$ and $\ell \geq 100\sqrt{k}$ the following statement is false: for $M = 3k^3\ell\, 2^{k\ell}$, whenever $\det_M$ has a circuit with $L$ gates, the polynomial $f_{k,\ell}$ has one with $5L$ gates.*

*Proof.* Suppose the statement holds for some such $k$ and $\ell$. By Inherited Result 23.5 the polynomial $\det_M$ has a circuit with $L = CM^{c_1}$ gates, so $f_{k,\ell}$ has one with at most $5CM^{c_1}$ gates. For $k \geq 4$ one has $\log(3k^3\ell) \leq 2 + 3\log k + \ell \leq k\ell$, hence $\log M \leq 2k\ell$ and $\log(5CM^{c_1}) \leq 2c_1 k\ell + \log(5C) \leq 3c_1 k\ell$ once $k\ell \geq \log(5C)/c_1$. By Proposition 16.8 every circuit for $f_{k,\ell}$ has more than $2^{\ell k^{3/2}/16}$ gates, so $3c_1 k\ell > \ell k^{3/2}/16$, that is $\sqrt{k} < 48c_1$. This is false for $k \geq k_0$ with $k_0 > \max(4, (48c_1)^2)$ and large enough that $k_0\ell \geq \log(5C)/c_1$. □

*Remark* 23.10. Proposition 23.9 is the exact form, in the flat case, of the statement that the proof fails for the determinant at the reduction of clause sums and nowhere else. It uses the lower bound for $f_{k,\ell}$, which is conditional on the flat condition, and it does not use the permanent at all.

**Proof Obligation 23.11.** *Identify exactly where the source proof invokes completeness rather than the algebraic form of the permanent. The dependency analysis above locates it in Inherited Result 14.7 for the flat case and in the clause-sum form of $\Phi$ for the contracting case, but it rests on the source's own reductions and on its Figure 1, and has not been checked against the Lean development, where the corresponding declarations should be traced.*

# Chapter 24

# Border Complexity and Limiting Representations

## 24.1 Approximation Without Exact Realization

Border complexity asks for circuits that compute a polynomial only in a limit. A circuit over the field of formal Laurent series $\mathbb{F}((\varepsilon))$ approximates $f$ if its output is $f + \varepsilon g$, where $g$ has coefficients in $\mathbb{F}[[\varepsilon]]$. The circuit may use constants with negative powers of $\varepsilon$, so that individual intermediate quantities become singular as $\varepsilon \to 0$, and only the final output is required to converge to $f$. Every exact circuit is an approximating circuit with $g = 0$, so that border complexity never exceeds exact complexity, and the interest of the notion is that it may be strictly smaller. For this reason a lower bound on border complexity is a stronger statement than a lower bound on exact complexity, and the source proves the stronger one.

The separation is therefore not only VNP $\not\subseteq$ VP but VNP $\not\subseteq \overline{\text{VP}}$, where $\overline{\text{VP}}$ is defined like VP with approximating circuits. The geometric framework suggests why the stronger statement should be reachable: the spectral obstruction is a property of the tensor that is represented, and an approximation of a tensor in the sense of limits is controlled when the representation has a bounded number of terms in each power of the parameter, which is the case here.

## 24.2 Formal Laurent Series

The formalism in the source is formal and not analytic. The parameter $\varepsilon$ is an indeterminate, the field is $\mathbb{F}((\varepsilon))$ of formal Laurent series, and no topology on $\mathbb{F}$ is used. Over an algebraically closed field, a polynomial is approximated with $L$ gates in this sense exactly when it lies in the Zariski closure of the polynomials with circuits of $L$ gates [3, 5], and over the complex numbers this closure is related to the closure in the Euclidean topology, but the source does not use any of these identifications and neither does this monograph. The first technical step in the source is to show that the constants may be taken to be Laurent polynomials, so that only finitely many powers of $\varepsilon$ occur in each constant. The step is a truncation, and it

is elementary enough to be proved here.

## 24.3 Singular Intermediate Quantities

The difficulty in passing from exact to approximating circuits is the one just mentioned: the negative powers of $\varepsilon$ in the constants. In the flat case the spectral argument counts the number of terms of the functionals that appear in the decomposition of the coefficient tensor. A functional with coefficients in $\mathbb{F}[\varepsilon, \varepsilon^{-1}]$ with exponents of $\varepsilon$ in $[-\Delta, \Delta]$ is a sum of $2\Delta + 1$ functionals over $\mathbb{F}$, one for each power. The $n$th power of such a functional has at most $2n\Delta + 1$ terms, so that the number of terms that the argument must cover grows, over $|V|$ parties, by a factor of $(2n\Delta + 1)^{|V|}$, a factor that is subexponential in the quantity $n$ at which the argument is applied. This is why the flat case survives approximation at the cost of a subexponential factor, and it is the content of the approximate version of the lemma on sums with few terms.

In the contracting case the problem is more serious, because an approximating circuit gives a program that is correct only up to multiples of $\varepsilon$, and its constants no longer lie in a ring in which the diagonal argument of Chapter 21 can be run. The source overcomes this by weakening the notion of accelerator to an approximate accelerator, whose constants lie in $R[\varepsilon, \varepsilon^{-1}]$ and whose values on plain instances are required only to lie in $\mathrm{sum}(z) + \varepsilon R[\varepsilon]$. The proof that such accelerators do not exist carries the program to larger scales while keeping the error a multiple of $\varepsilon$, substituting $\varepsilon \mapsto \varepsilon^{r}$ at each round so that the error from one scale is absorbed at the next, and finally extracts the coefficient of $\varepsilon^{0}$. The extraction guesses, for each named power of a constant, the number of times it is used and the index of the term of that power, and keeps a counter of the exponents of $\varepsilon$ modulo a power of two chosen larger than the possible sum. The result is an exact accelerator of polynomial size and fixed exponent, which Theorem K.6 excludes.

## 24.4 Persistence of Obstruction Under Limits

The obstruction therefore persists under limits for a structural reason. In each case the parameter $\varepsilon$ adds to the argument a bounded amount of bookkeeping, the number of terms in each power and the number of rounds, and the bookkeeping is absorbed by the slack in the exponents. The flat bound has slack in the choice of the parameters $k$ and $\ell$, and the contracting bound has slack in the factor by which the running time is allowed to exceed the minimum at each scale. The statement of the source, that the principal bound holds for approximating circuits with the same exponent, is consistent with this picture: the exponent is not changed, because the added terms are subexponential in the quantities that set the exponent.

The consequence for determinants is the source's Remark N.6. Over a field of characteristic different from two, a determinant of an affine matrix with entries in $\mathbb{F}((\varepsilon))$ and size $N$ that approximates $\mathrm{perm}_m$ yields an approximating circuit of size polynomial in $N$ and $m$,

by Berkowitz's construction applied over $\mathbb{F}((\varepsilon))$, and the border lower bound then bounds $N$ from below by $m^{c'(\log m)^{1/3}}$ exactly as in Corollary 23.8. The source records that this remark is the one numbered statement of the manuscript without a Lean equivalent.

## 24.5 Connections to Geometric Complexity Theory

The statement that the permanent is not approximated by determinants of polynomial size is a form of the conjecture of Mulmuley and Sohoni, expressed in the language of orbit closures [23, 24]. The source observes that for matrices of polynomial size over the complex numbers this was conjectured by them, and that its own result is the subexponential bound $m^{c(\log m)^{1/3}}$ and not the polynomial one. The distinction should be kept in view: the monograph neither claims nor needs the conjecture, and the geometric structure that it develops, spectral points and pseudometrics on tensors, is of a different kind from the representation-theoretic obstructions of geometric complexity theory. The two approaches share the feature that they study the closure of a set of low-complexity objects and ask whether a target lies outside it, and this is the reason for the placement of the chapter in Part VIII, where the boundaries of the theorem are mapped.

## 24.6 The Three Boundaries Together

The three chapters of this part examine the preservation of obstruction under three transformations. Boolean quotienting asks whether obstruction survives the loss of formal distinctions, and the answer of Chapter 22 is that it does not, because the extraction operators have no descent to the quotient. Polynomial projection asks whether an obstructed structure can be realized through a tractable family, and the answer of Chapter 23 is that it cannot for the determinant, so that the determinant serves as a negative control: it shares the monomial support of the permanent and escapes at the single step where completeness is used. Formal degeneration asks whether obstruction survives limiting representations, and the answer of this chapter, inherited from the source, is that it does, with the same rate. Obstruction is therefore destroyed by one transformation, blocked from transfer by a second, and preserved under a third, and the conditions that distinguish these cases are the material from which the general theory of Part XIII is to be built. The graph-cut example of the later parts will illustrate these conditions using the cut-rank identity of Chapter 9 and is not anticipated here.

## 24.7 Formal Statements

**Definition 24.1** (Approximating Circuit)**.** A circuit over $\mathbb{F}((\varepsilon))$ approximates a polynomial $f$ over $\mathbb{F}$ if it computes a polynomial of the form $q = f + \varepsilon g$ with all coefficients of $g$ in $\mathbb{F}[[\varepsilon]]$. We fix one such $q$ and call it the output.

**Definition 24.2** (Border Circuit Complexity)**.** $\underline{L}_{\mathbb{F}}(f) = \inf\{L_{\mathbb{F}((\varepsilon))}(q) : q = f + \varepsilon g,\ g \in \mathbb{F}[[\varepsilon]][x]\}$. The class $\overline{\mathrm{VP}}$ is defined like VP with $\underline{L}_{\mathbb{F}}$ in place of $L_{\mathbb{F}}$, and polynomial degree as before.

**Proposition 24.3** (Border Complexity Does Not Exceed Exact Complexity)**.** $\underline{L}_{\mathbb{F}}(f) \le L_{\mathbb{F}}(f)$ *for every* $f$*, and* $\mathrm{VP} \subseteq \overline{\mathrm{VP}}$.

*Proof.* An exact circuit for $f$ over $\mathbb{F}$ is a circuit over $\mathbb{F}((\varepsilon))$ computing $q = f$, which has the required form with $g = 0$. The inclusion of the classes follows since the degree condition is unchanged. □

**Proposition 24.4** (Truncation of Constants)**.** *Let* $f$ *be a polynomial over* $\mathbb{F}$ *and let a circuit with* $L$ *gates over* $\mathbb{F}((\varepsilon))$ *approximate* $f$*, with constants* $c_1, \dots, c_r$ *of order at least* $-\Delta$ *in* $\varepsilon$*, where* $\Delta \ge 1$*. Let* $N = 2^L\Delta$ *and let* $c_i'$ *be the sum of the terms of* $c_i$ *with exponent below* $N$*. The circuit with constants* $c_i'$ *has* $L$ *gates and computes a polynomial in* $f + \varepsilon\,\mathbb{F}[\varepsilon][x]$.

*Proof.* Unfold the circuit as a polynomial $P(c, x)$ with integer coefficients in the constants and the variables. By induction along the circuit the polynomial computed at the $j$th gate has degree at most $2^j$, since a sum does not raise the degree and a product at most adds the degrees of its factors, so every monomial of $P$ has at most $d \le 2^L$ factors that are constants. Let $u_1 \cdots u_d$ be such a product of constants and $u_i'$ the corresponding truncations. Then $u_i$ and $u_i'$ have order at least $-\Delta$, and $u_i - u_i'$ has order at least $N$. The telescoping identity

$$u_1 \cdots u_d - u_1' \cdots u_d' = \sum_{j=1}^{d} u_1' \cdots u_{j-1}'(u_j - u_j')u_{j+1} \cdots u_d$$

shows that the difference has order at least $N - (d-1)\Delta \ge N - (2^L - 1)\Delta = \Delta \ge 1$. Hence $P(c', x) - P(c, x)$ has all coefficients in $\varepsilon\mathbb{F}[[\varepsilon]]$, and since $P(c, x) = f + \varepsilon g$ with $g$ over $\mathbb{F}[[\varepsilon]]$, also $P(c', x) - f$ has coefficients in $\varepsilon\mathbb{F}[[\varepsilon]]$. These coefficients also lie in $\mathbb{F}[\varepsilon, \varepsilon^{-1}]$, because the $c_i'$ are Laurent polynomials, and the intersection of $\varepsilon\mathbb{F}[[\varepsilon]]$ with $\mathbb{F}[\varepsilon, \varepsilon^{-1}]$ is $\varepsilon\mathbb{F}[\varepsilon]$. □

**Inherited Result 24.5** (Constants That Are Laurent Polynomials)**.** *Let* $f$ *have integer coefficients and let* $\operatorname{char}\mathbb{F} = p$*. If a circuit with* $L$ *gates over* $\mathbb{F}((\varepsilon))$ *approximates* $f$*, then some circuit with* $L$ *gates over the ring* $K[\varepsilon, \varepsilon^{-1}]$ *computes a polynomial in* $f + \varepsilon K[\varepsilon][x]$*. This is Lemma N.2 of the manuscript. The truncation step is Proposition 24.4, and the passage to the coefficient ring* $K$ *repeats the argument of Lemma M.1 on the coefficients, which is not reproved here.*

**Proposition 24.6** (Support of Powers)**.** *If a Laurent polynomial* $p$ *over a field has all exponents of* $\varepsilon$ *in* $[-\Delta, \Delta]$*, then* $p^n$ *has all exponents in* $[-n\Delta, n\Delta]$ *and therefore at most* $2n\Delta + 1$ *terms.*

*Proof.* The exponents of a product of $n$ terms of $p$ are sums of $n$ integers in $[-\Delta, \Delta]$, so lie in $[-n\Delta, n\Delta]$, an interval with $2n\Delta + 1$ integers, and a Laurent polynomial has at most one term for each exponent. □

**Inherited Result 24.7** (Sums With Few Terms, Approximate Version)**.** *Lemma E.7 of the manuscript remains true if the functionals have coefficients in* $\mathbb{F}[\varepsilon, \varepsilon^{-1}]$*, no negative power of* $\varepsilon$ *occurs in* $\sum_j f_j(T_j)$*, and* $t$ *is the coefficient of* $\varepsilon^0$ *in this sum. The proof follows that of Lemma E.7 with exponents in* $[-\Delta, \Delta]$*, using Proposition 24.6 to cover the coefficient of* $\varepsilon^0$ *in the* $n$*th power by* $(2n\Delta + 1)^{|V|}$ *copies of the same tensor, a subexponential number. This is Lemma N.3, and with it Proposition E.10 and Theorem E.11 hold for approximating circuits in the flat case.*

**Inherited Result 24.8** (Absence of Approximate Accelerators)**.** *An approximate accelerator over an* $A$*-algebra* $R$ *is defined as in Definition 20.7, except that its constants lie in* $R[\varepsilon, \varepsilon^{-1}]$ *and that its value on a plain instance* $z$ *is only required to lie in* $\operatorname{sum}(z) + \varepsilon R[\varepsilon]$*. Then Theorem K.6 holds for approximate accelerators over every nonzero finitely generated* $A$*-algebra. This is Proposition N.4 of the manuscript. Its proof carries the program through the rounds of Inherited Result 20.9 with the substitutions* $\varepsilon \mapsto \varepsilon^r$*, then extracts the coefficient of* $\varepsilon^0$ *by guessing the usage counts and terms of the named powers and tracking the exponents modulo a power of two.*

**Inherited Result 24.9** (Border Lower Bound)**.** *Under the quantifiers of Chapter 6, there are* $c > 0$ *and thresholds* $m_0$ *such that for* $m \geq m_0$*,* $\underline{L}_{\mathbb{F}}(\mathrm{HC}_m) > m^{c(\log m)^{1/3}}$*, and, for* $\operatorname{char} \mathbb{F} \neq 2$*, the same holds for* $\mathrm{perm}_m$*. This is Theorem N.5 of the manuscript. Its proof applies Inherited Result 24.5, treats the flat case by Inherited Result 24.7 and the contracting case by Inherited Result 24.8 with the calculation of Theorem 21.7.*

*Remark* 24.10. Since $m^{c(\log m)^{1/3}} = 2^{c(\log m)^{4/3}}$, Inherited Result 24.9 is the statement $\underline{L}_{\mathbb{F}}(\mathrm{HC}_m) \geq 2^{\Omega((\log m)^{4/3})}$, the same rate as the exact bound. The persistence of the bound under approximation is therefore a statement about the rate and not only about superpolynomiality, and it is a statement about formal Laurent series: no convergence of $\sum a_j \varepsilon^j$ in any topology on $\mathbb{F}$ is used, and the cancellation of negative powers in the output is an identity in $\mathbb{F}[[\varepsilon]]$.

**Corollary 24.11** (Exact Lower Bound from the Border Bound)**.** *Inherited Result 24.9 implies the corresponding lower bound for* $L_{\mathbb{F}}(\mathrm{HC}_m)$ *and* $L_{\mathbb{F}}(\mathrm{perm}_m)$*.*

*Proof.* By Proposition 24.3, $L_{\mathbb{F}} \geq \underline{L}_{\mathbb{F}}$. □

**Corollary 24.12** (VNP Is Not Contained in the Closure of VP)**.** *Assume Inherited Result 24.9. Then* $\mathrm{VNP}_{\mathbb{F}} \not\subseteq \overline{\mathrm{VP}}_{\mathbb{F}}$*.*

*Proof.* The family $(\mathrm{HC}_m)$ lies in VNP. If it lay in $\overline{\mathrm{VP}}$, then $\underline{L}_{\mathbb{F}}(\mathrm{HC}_m)$ would be bounded by a polynomial in $m$, whereas $m^{c(\log m)^{1/3}}$ exceeds every polynomial for large $m$. □

**Corollary 24.13** (Approximate Determinantal Complexity of the Permanent)**.** *Assume Inherited Result 24.9 with* $\operatorname{char} \mathbb{F} \neq 2$*. There are* $c' > 0$ *and* $m_1$ *such that for* $m \geq m_1$*, if* $\mathrm{perm}_m$ *is approximated, in the sense of Definition 24.1, by* $\det(A)$ *for an* $N \times N$ *matrix* $A$ *with entries affine over* $\mathbb{F}((\varepsilon))$*, then* $N \geq m^{c'(\log m)^{1/3}}$*.*

*Proof.* The argument of Proposition 23.7 applies over $\mathbb{F}((\varepsilon))$ with $\underline{L}$ in place of $L$, since Inherited Result 23.5 holds over every field, in particular over $\mathbb{F}((\varepsilon))$, and the entries of $A$ are computed from the variables with at most $5m^2$ entries each. The conclusion follows as in Corollary 23.8. This is Remark N.6 of the manuscript, which is the one numbered statement there without a Lean equivalent, and the argument here shares that status as a consequence of Inherited Result 24.9 and Berkowitz's theorem. □

**Proof Obligation 24.14.** *Reconstruct the approximation-transfer arguments of Appendix N of the source, in particular the construction of the approximate accelerators in Proposition N.4, including the substitution rounds and the extraction of the coefficient of $\varepsilon^0$. The printed proof describes the programs and accounts for their sizes to the level of exponents, and states that the programs perform as claimed only in the Lean development, so that a reconstruction must reproduce the programs or audit the formal declarations.*

# Part IX

# From Existential Proof to Operational Knowledge

# Chapter 25

# The Limits of Nonconstructive Achievement

## 25.1 Existential Lower Bounds

The principal result of the source manuscript has the logical form of an eventual statement. It asserts that there are an absolute constant $c > 0$ and a threshold $m_0$, depending on the characteristic of the field, such that for every $m \geq m_0$ the circuit complexity of the Hamiltonian cycle polynomial exceeds $m^{c(\log m)^{1/3}}$. A statement of this form is a genuine mathematical achievement and it has a definite content: it excludes polynomial-size circuits for every polynomial bound. It does not, however, say how large $m_0$ is, and it does not assert that any particular value of $c$ can be written down. The distinction between what is asserted and what would be needed to use the assertion in a computation is the subject of this chapter, and it is a distinction that applies to a great deal of completed mathematics besides this theorem.

The quantifier structure matters because a proof of $\exists m_0 \forall m \geq m_0 \, R(m)$ may proceed by a route that never produces a number. In the flat case of the present argument the route is explicit enough that the threshold can in principle be read off from the stated inequalities, which are conditions on the integer $r$ of Theorem 16.9, although no value is computed in this monograph. In the contracting case the threshold depends on the constants of the cutting construction and of the accelerator, which the source does not make explicit, and the monograph says so in the remark following Theorem 21.7.

## 25.2 The Unknown Threshold

That an eventual statement is true does not imply that its threshold can be computed, and this can be shown in a form that does not depend on any particular theorem. For a Turing machine $e$, consider the statement that $e$ does not halt at exactly step $m$ for every sufficiently large $m$. Each such statement is true, since a machine halts at most once, and a single proof by cases establishes all of them uniformly. Yet there is no computable function

that, given $e$, returns a valid threshold, for such a function would decide the halting problem. The example is not claimed to resemble the principal theorem. It shows that the form of statement, by itself, guarantees nothing about computability of the threshold, so that this must be established separately for each result.

The same point applies to the alternative between the flat and the contracting regimes. The source proves that one of the two holds for every field and proves the lower bound in each case. The lower bound follows by cases, and a proof by cases does not require a procedure that decides which case obtains. The theorem is therefore established without any account of which regime is realized, a feature that is mathematically unobjectionable and that has the consequence, stated in the final section of this chapter, that the proof delivers an existence statement whose explanation is split.

## 25.3 Nonexplicit Obstruction Functionals

The obstruction in the geometric argument is carried by spectral points, which are functionals on tensors satisfying the axioms of Chapter 8. The existence of spectral points below a given upper functional is proved in the source by Zorn's lemma, and the duality that converts a bound on all spectral points into a bound on covers is proved from that existence. The functionals that the argument uses are therefore not exhibited. The argument needs only that they exist and that they pull back along the distance functions of Chapter 9 to give the exponents $d_\varphi$, and it proceeds by choosing one for each placement of the ends and hubs. A reader who wished to verify the lower bound numerically would find that the obstruction functional is not a computable object in the sense of Chapter 25 of the outline, although the inequalities it satisfies are explicit. This is a second kind of nonconstructive element, distinct from the unknown threshold, and it is recorded here because it affects the operational accessibility of the result: the certificate, if the inherited results are granted, is a proof that no small circuit exists, and not a procedure that exhibits, for a given small circuit, a point at which it fails.

## 25.4 Four Levels of Mathematical Accessibility

It is useful to distinguish four levels at which a mathematical statement may be available to an investigator. At the propositional level the statement has been established within an accepted framework, that is, a valid certificate exists. At the certificational level a proof object is in hand and the verification relation can be run on it within the investigator's resources. At the witness level, for a statement of the form $\exists w\, R(w)$, a procedure is available that produces a witness. At the operational level that procedure, or the certificate check, can be executed within the resources the investigator actually has. These levels are not presented as an unconditional hierarchy of logical implications. Their relationships depend on the proof system, on the type of statement, on the representation of witnesses, and on the computational environment, and the formal statements below record the relations that

can be proved and the separations that can be exhibited, without asserting a linear order.

The environment-relative cost of Chapter 1 gives the quantitative form of the second and fourth levels. If an environment $K'$ makes more certificates accessible than $K$, with no change in the cost of any certificate, then the effective reconstruction cost cannot increase, and it can fall from infinity to a finite value. This describes precisely how mathematical knowledge can become operationally easier without any change in mathematical truth, for example when a formalization is released, when a library is published, or when a toolchain is made available. The converse also holds: when access is withdrawn, the same truth becomes operationally unavailable. These observations are elementary and their proofs are short, but they supply the vocabulary in which the later chapters on verification and on dispossession are stated.

## 25.5 The Residual Difficulty of Completed Mathematics

The thesis of Chapter 1 is that completed mathematics replaces search by verification, and that this replacement is conditional on the possession of a certificate. The case study of the principal result qualifies the thesis in two ways. The first is that a certificate may establish an eventual statement without a threshold, so that a user of the theorem cannot, from the certificate alone, decide for a given $m$ whether the bound applies. The second is that parts of the certificate in this case are inherited, so that its checkability depends on the formal development of the source and on the correspondence between that development and the printed statements, a matter taken up in Part X. The difficulty that remains after a result has been proved is therefore of several kinds: the difficulty of extracting numbers, the difficulty of identifying which of several cases holds, the difficulty of auditing the certificate, and the difficulty of maintaining the artifact. None of these negates the achievement. Each is a cost that the achievement leaves to those who use it, and the accounting of these costs is the subject of the next chapter.

## 25.6 Formal Statements

**Definition 25.1** (Accessibility Profile)**.** Let $V$ be a verification relation and $K$ a knowledge environment, as in Chapter 1. For a statement $T$ and a resource bound $R$, the accessibility profile $\mathcal{A}_R(T) = (A_P, A_C, A_W, A_O) \in \{0,1\}^4$ is defined by $A_P = 1$ iff $\mathcal{C}(T) \neq \varnothing$, and $A_C = 1$ iff $C_K(T) < \infty$. For $T$ of the form $\exists w \in W\, R_T(w)$, $A_W = 1$ iff $K$ contains a procedure that computes some $w \in W_T$, and $A_O = 1$ iff $C_K(T) \leq R$ and the procedure of $A_W$ runs within $R$. Each component is relative to the stated $V$, $K$ and $R$.

**Definition 25.2** (Existential Witness Problem)**.** For $T : \exists w \in W\, R_T(w)$ let $W_T = \{w : R_T(w)\}$. A proof of $T$ establishes $W_T \neq \varnothing$. It does not necessarily supply a computable selection $s(T) \in W_T$, and a family $(T_e)_e$ is said to have computable witnesses if some computable $s$ satisfies $s(e) \in W_{T_e}$ for all $e$.

**Proposition 25.3** (Relations Among the Levels)**.** *For every $T$, $A_C = 1$ implies $A_P = 1$, and $A_O = 1$ implies $A_C = 1$ and $A_W = 1$. If $R_T$ is decidable by a procedure available in $K$, then $A_W = 1$ implies $A_C = 1$. The converses of the first two implications, and of the last, can fail.*

*Proof.* If $C_K(T) < \infty$ the infimum defining it is over a nonempty set of valid certificates, so $\mathcal{C}(T) \neq \varnothing$. The second implication is part of the definition of $A_O$. If $R_T$ is decidable in $K$ and a procedure computes $w \in W_T$, then $w$ is a certificate that is accessible and checkable by running the decision procedure, so $C_K(T) < \infty$. The first converse fails by Proposition 1.9, since existence of a certificate does not imply that one is accessible. The family of Proposition 25.4 shows that $A_C$ does not imply $A_W$ for families, and an exponential-time brute-force procedure for a statement with a verifiable witness shows that $A_W$ does not imply $A_O$ for modest $R$. □

**Proposition 25.4** (Existential and Operational Separation)**.** *For a Turing machine $e$ let $T_e$ be the statement $\exists m_0 \, \forall m \geq m_0$ ($e$ does not halt at step exactly $m$). Every $T_e$ is true, and provably so by a single uniform argument, but there is no computable function $s$ with $s(e)$ a valid threshold for $T_e$ for all $e$.*

*Proof.* If $e$ halts, it halts at exactly one step $h$ and $m_0 = h + 1$ is a valid threshold. If it never halts, $m_0 = 0$ is valid. Suppose a computable $s$ returned a valid threshold for each $e$. Then $e$ halts if and only if it halts within $s(e)$ steps, because a halt at any step $m \geq s(e)$ is excluded, and this is decided by simulating $e$ for $s(e)$ steps, so the halting problem would be decidable, which it is not. □

*Remark* 25.5. Proposition 25.4 shows that a true eventual statement with a uniform proof need not have a computable threshold. It is a statement about the logical form and is not a claim about the principal theorem, whose threshold is explicit in the flat case in principle and unavailable in the contracting case.

**Proposition 25.6** (Monotonicity of Reconstruction Cost in the Environment)**.** *Let $K \subseteq K'$ mean that every certificate accessible in $K$ is accessible in $K'$ at no greater access cost. Then $C_{K'}(T) \leq C_K(T)$ for every $T$. If $C_K(T) = \infty$ and $K'$ makes some valid certificate accessible, then $C_{K'}(T) < \infty$.*

*Proof.* The infimum defining $C_{K'}(T)$ is over a set containing every certificate of the set defining $C_K(T)$, each with a cost that is no greater, so the infimum is no greater. If $C_K(T) = \infty$ the inequality is trivial, and a single accessible valid certificate in $K'$ has finite cost, since the verification cost of a certificate is finite by the definition of a verification relation, so the infimum is finite. □

*Remark* 25.7. Proposition 25.6 changes the cost without changing $\mathcal{C}(T)$, so it changes operational accessibility without any change in mathematical truth. The inequality can be strict, and the reverse inclusion $K' \subseteq K$ gives the reverse inequality.

**Proposition 25.8** (Proof by Cases Without Selection)**.** *Let* $\mathsf{F}$*,* $\mathsf{C}$ *and* $\mathsf{L}$ *be statements and suppose* $\mathsf{F} \vee \mathsf{C}$*,* $\mathsf{F} \Rightarrow \mathsf{L}$ *and* $\mathsf{C} \Rightarrow \mathsf{L}$ *are established. Then* $\mathsf{L}$ *is established, and the established premises do not by themselves determine which of* $\mathsf{F}$*,* $\mathsf{C}$ *holds.*

*Proof.* The first claim is disjunction elimination. For the second, the premises are also satisfied when both $\mathsf{F}$ and $\mathsf{C}$ hold, and when exactly one holds in either of the two ways, so they cannot discriminate among these cases. □

**Application 25.9.** *Inherited Results 13.6, together with Theorems 16.9 and 21.7, have the form of Proposition 25.8 with* $\mathsf{L}$ *the lower bound for the permanent. The principal theorem is thereby established without any account of which regime is realized for a given field, and the threshold* $m_0$ *is explicit only in the flat case.*

*Example* 25.10 (Polynomial Time Without Practicality)*.* An algorithm with running time $T(n) = 10^{100} n^{200}$ is polynomial in the sense of the asymptotic definition. At $n = 2$ it requires $10^{100} \cdot 2^{200} \approx 1.6 \times 10^{160}$ steps, a number that exceeds the commonly cited estimate of the number of atoms in the observable universe, of order $10^{80}$, by eighty orders of magnitude. Hence an existential proof of polynomial-time solvability, an explicit algorithm of polynomial running time, and an algorithm that can be run on inputs of interest are three distinct properties, in the order of the accessibility profile of Definition 25.1. Separately, a proof that every problem in $\mathsf{NP}$ has a polynomial-time algorithm would concern computational hardness assumptions and would leave unaffected schemes whose security is information-theoretic, such as the one-time pad, which do not rest on such assumptions.

**Proof Obligation 25.11.** *Compute an explicit* $m_0$ *for the flat case from the inequalities in the proof of Theorem 16.9, and determine whether the contracting case admits any explicit threshold. The first is a finite calculation from the stated conditions on* $r$*. The second requires making explicit the constants of the cutting construction, which the source does not do.*

# Chapter 26

# Mathematical Knowledge as Amortized Computation

## 26.1 The Cost of Discovery

Discovery expenditure is the cost of producing a certificate in the first place. It is the most visible cost, because it is the one that history records in years and in the number of capable researchers who attempted the problem, and it is also the least well defined, because it may denote the historical expenditure, the least expenditure among known procedures, or a hypothetical minimum, as Chapter 2 observes. For the present purposes it enters the accounting as a single nonnegative constant $D$ paid once. Its size is relevant to the decision whether to develop a result, and it is irrelevant to the cost of using the result after it exists, which is the reason that amortization can make a result inexpensive per use even when its discovery was very costly.

## 26.2 The Cost of Verification

Verification expenditure is the cost of confirming that a certificate is valid. For a formal proof it is the cost of running a checking kernel on a proof object, and for an informal proof it is the cost of reading and checking it, which is a quantity that depends on the reader. It is paid once by each party that does not rely on the verification of another, so that the number of independent verifications is itself a parameter of the accounting. The monograph treats verification as a cost that may be shared, in which case it enters $V$ once, or repeated, in which case it enters the application costs. The distinction becomes important in Part X, where the question is how much of the verification of the principal result can be inherited from the Lean development without being repeated.

## 26.3 The Cost of Application

Application expenditure is the cost of using a result in a particular instance, for example of deploying a lower bound in a calculation, or of citing the theorem in a proof. It is paid at every use, and it may vary from one use to the next, so that the sequence $A_1, A_2, \ldots$ is part of the data. A result with a high application cost is difficult to use even if it is cheap to verify, and the principal theorem is an illustration: its use requires knowing the threshold, which the previous chapter showed to be unavailable in general.

## 26.4 Amortization Across Repeated Use

The mean cost per use is the sum of the fixed costs divided by the number of uses, the maintenance cost per use, and the mean application cost, as in Chapter 2. The identity gives the limit when maintenance grows sublinearly and the mean application cost converges. The decisive question is not the limit but the break-even point, the number of uses after which reuse is cheaper than repeated rediscovery. If independent rediscovery costs $d$ per application and reuse costs $a$ per application together with a fixed establishment cost $F$ and maintenance $M(N)$, reuse is cheaper exactly when $N(d - a) > F + M(N)$. The advantage of accumulated knowledge therefore depends not only on the difficulty of discovery, through $d$, but on the durability and accessibility of the certificates, through $M(N)$ and through any access cost that raises $a$. If maintenance grows linearly at a rate $\mu$ that exceeds the per-use saving $d - a$, the advantage never appears, and if access is restricted so that the effective $a$ rises above $d - \mu$, it disappears. This is a mathematical statement about an accounting model, and it requires no sociological assumption, although it identifies the quantities that a sociological account of the loss of access would have to measure.

## 26.5 Mathematical Civilization as Accumulated Certificates

A body of mathematics is an accumulation of certificates, definitions and techniques, each of which has been paid for once and is used many times. The accounting of this chapter applies to each element and to the aggregate, and the aggregate has the property that reuse of one element reduces the application cost of others, because a theorem is typically proved from earlier ones. The general principle is that the per-use cost of a body of knowledge is low where the artifacts are durable, accessible and maintained, and that these are conditions on the environment and not on the truth of the results. The monograph does not offer an estimate of any actual institution’s costs. It offers the vocabulary and the elementary propositions in which such an estimate would be stated, and it records that claims about actual infrastructure require evidence that the propositions do not supply.

## 26.6 Explanation as Further Compression

A proof eliminates the necessity of rediscovering a particular result. An explanation may in addition eliminate the necessity of independently discovering many related results, and this is the sense in which explanatory compression differs from discovery compression. For a finite family of statements derivable from a background theory, the gain from introducing a structural result is the total reduction in derivational complexity over the family, net of the cost of establishing and maintaining the result, as in the hypothesis of Chapter 2. The functional is not a measure of proof length alone. It measures the downstream benefit of a reusable structure, and it can be negative when a structure is more expensive to maintain than the derivations it shortens.

The spectral geometry of the preceding parts is a candidate for this criterion. Its explanatory value would reside not only in the separation of VP from VNP but in the extent to which its constructions reduce the difficulty of investigating other forms of computational obstruction, and that is a statement about a family of future derivations that this monograph cannot evaluate. The same functional supplies a criterion for the original contributions of the later parts, which are to be judged by the family of problems on which they reduce derivational complexity and not by their length or novelty.

## 26.7 Formal Statements

**Definition 26.1** (Knowledge Expenditure)**.** For a statement $T$, let $C_D(T)$ be the discovery expenditure, $C_V(T, \pi)$ the verification expenditure of a certificate $\pi$, $C_I$ the infrastructure expenditure, $C_M(N)$ the cumulative maintenance expenditure over $N$ uses, and $A_i \geq 0$ the application expenditure of the $i$th use. All are nonnegative reals, and the cumulative expenditure after $N$ uses is $C(N) = C_D + C_V + C_I + C_M(N) + \sum_{i=1}^{N} A_i$.

**Proposition 26.2** (Break-Even Against Rediscovery)**.** *Suppose rediscovery costs $d$ per use, so that $N$ uses cost $Nd$, and reuse costs a fixed establishment $F \geq 0$, cumulative maintenance $M(N) \geq 0$ and $a$ per use. Then reuse costs strictly less than rediscovery after $N$ uses if and only if $N(d - a) > F + M(N)$. If $M(N) = \mu N$ with $\mu \geq 0$, this holds for some $N$ if and only if $d > a + \mu$, and then for every $N > F/(d - a - \mu)$.*

*Proof.* Reuse costs $F + M(N) + Na$ and rediscovery costs $Nd$, so reuse is cheaper iff $F + M(N) + Na < Nd$, which is the stated condition. With $M(N) = \mu N$ the condition is $N(d - a - \mu) > F$. If $d - a - \mu \leq 0$ the left side is at most zero while $F \geq 0$, so it fails for every $N \geq 1$. If $d - a - \mu > 0$ it holds exactly for $N > F/(d - a - \mu)$. □

**Corollary 26.3** (Loss of Advantage Under Restricted Access)**.** *If access to the infrastructure adds a cost $\kappa \geq 0$ to every use, replacing $a$ by $a + \kappa$, then with linear maintenance reuse is never cheaper than rediscovery once $\kappa \geq d - a - \mu$. If the certificate becomes inaccessible, in the sense that $C_K(T) = \infty$, reuse is impossible and the cost of rediscovery is the only finite option.*

*Proof.* The first claim is Proposition 26.2 with $a + \kappa$ in place of $a$, since $d > a + \kappa + \mu$ fails when $\kappa \geq d - a - \mu$. The second holds because the cost of reuse is the infimum over an empty set, which is infinite by definition. □

**Proposition 26.4** (Break-Even for a Simplified Derivation)**.** *Suppose an alternative derivation costs $D' > 0$ to develop and saves $\delta > 0$ in each subsequent use. Then it reduces cumulative expenditure after $N$ uses if and only if $N > D'/\delta$.*

*Proof.* The cumulative saving is $N\delta$ against the one-time cost $D'$, and $N\delta > D'$ iff $N > D'/\delta$. □

*Remark* 26.5. Proposition 26.4 gives a formal justification for continuing to simplify mathematics after its initial proof, when the number of future uses is large. It says nothing about whether the number of uses will in fact be large, which is an empirical matter.

**Definition 26.6** (Explanatory Compression Ratio)**.** For two valid proofs $\pi_1, \pi_2$ of the same statement, $\Gamma(\pi_1, \pi_2) = |\pi_1|/|\pi_2|$, and, with expenditures in place of lengths, $\Gamma_C(\pi_1, \pi_2) = \dfrac{C_V(\pi_1) + C_A(\pi_1)}{C_V(\pi_2) + C_A(\pi_2)}$, whenever the denominators are positive.

**Proposition 26.7** (Multiplicativity of the Ratio)**.** *For valid proofs $\pi_1, \pi_2, \pi_3$ of the same statement with positive lengths, $\Gamma(\pi_1, \pi_3) = \Gamma(\pi_1, \pi_2)\,\Gamma(\pi_2, \pi_3)$, and $\Gamma(\pi_1, \pi_2) > 1$ if and only if $\pi_2$ is strictly shorter than $\pi_1$. The same holds for $\Gamma_C$.*

*Proof.* The product $\frac{|\pi_1|}{|\pi_2|} \cdot \frac{|\pi_2|}{|\pi_3|}$ equals $\frac{|\pi_1|}{|\pi_3|}$ and a ratio of positive numbers exceeds one iff the numerator is larger. The same algebra applies to the sums of expenditures. □

**Definition 26.8** (Net Explanatory Gain)**.** Let $B$ be a background theory, $S$ a structural result, $\mathcal{F}$ a finite family of statements, $M < \infty$ a budget and $C_B(S) \geq 0$ the cost of establishing and incorporating $S$. With $K_B^{(M)}$ as in Chapter 2, $G_B(S; \mathcal{F}) = \sum_{T \in \mathcal{F}} [K_B^{(M)}(T) - K_{B \cup \{S\}}^{(M)}(T)] - C_B(S)$.

**Proposition 26.9** (Properties of the Gain)**.** *Each summand in $G_B(S; \mathcal{F})$ is nonnegative. Consequently, for $\mathcal{F} \subseteq \mathcal{F}'$, $G_B(S; \mathcal{F}) \leq G_B(S; \mathcal{F}')$. For disjoint families $\mathcal{F}_1, \mathcal{F}_2$, $G_B(S; \mathcal{F}_1 \cup \mathcal{F}_2) = G_B(S; \mathcal{F}_1) + G_B(S; \mathcal{F}_2) + C_B(S)$. If each statement of a family of size $n$ has its derivational complexity reduced by at least $\delta > 0$, then $G_B(S; \mathcal{F}) \geq n\delta - C_B(S)$, which is positive when $n > C_B(S)/\delta$.*

*Proof.* By Proposition 2.3, $K_{B \cup \{S\}} \leq K_B$, and truncation at $M$ preserves the order, so every summand is nonnegative. Adding terms to the family adds nonnegative summands while $C_B(S)$ is fixed, which gives monotonicity. For disjoint families the sums add and the cost term appears once on the left and twice on the right, which gives the stated correction. The last claim follows by bounding the sum below by $n\delta$. □

**Hypothesis 26.10** (Explanatory Reach of Spectral Geometry)**.** *There are families $\mathcal{F}$ of statements about computational obstruction, other than the separation of* VP *from* VNP*, for*

*which the gain $G_B(S;\mathcal{F})$ of introducing the structure $S$ of Chapters 8 to 13 is positive. The hypothesis is a conditional model. It requires a specification of $B$, $\mathcal{F}$ and the cost measure, and the monograph supplies no evidence for it beyond the structure of the argument.*

**Proof Obligation 26.11.** *Claims about the maintenance cost of any actual mathematical infrastructure, the costs of access to computational infrastructure, and the explanatory reach of spectral geometry require empirical evidence. The propositions of this chapter are conditional on the stated accounting model and do not supply such evidence.*

# Part X

# Formal Verification and Evidentiary Independence

# Chapter 27

# Proof Objects and Verification Architectures

## 27.1 The Distinction Between an Assertion and a Certificate

An assertion that a theorem has been proved, and a certificate that can be checked, are different objects, and the whole of this part turns on keeping them apart. The source manuscript asserts that its proofs have been formalized and verified in Lean, and it provides a development that can in principle be rebuilt. What a reader obtains from the assertion is a claim about a process. What a reader obtains from the development is an artifact on which a checking relation can be run. The first is evidence of a kind that depends on the credibility of the party who makes it, and the second is evidence that can be evaluated by anyone with the means to run the checker, subject to the qualifications of this chapter. The monograph records, throughout, the status of each of its statements in these terms, and this chapter applies the same discipline to the verification of the principal result.

The scope of what follows is limited by the material. The facts about the Lean development that are used below are those that the source manuscript reports about itself in its Section 3 and its Appendix Q: the toolchain and library versions, the axiom footprint, the checks performed, and the differences between the printed text and the formalization. They are recorded here as reports. None of them has been reproduced in this monograph, and the audit status vectors of the formal statements record this explicitly.

## 27.2 Formal Proof Systems

A formal proof system is determined by a language, a checking kernel, a set of admitted axioms and a body of library definitions on which statements rely. A certificate in such a system is a term or a script that the kernel accepts as establishing a formal statement from the axioms and the library. The kernel is a small program whose correctness is a different matter from the correctness of any proof it accepts, and the library is a large body of definitions and theorems whose own fidelity to the concepts they name is a further

dependency. A formal certificate is therefore conditional on four things: the soundness of the kernel, the consistency of the admitted axioms, the fidelity of the library definitions that the statement uses, and the fidelity of the translation from the intended statement into the formal language. The first two are the concern of the foundations of the proof assistant. The last two are the concern of whoever relies on a particular formalization, and they are the subject of the following sections.

## 27.3 The Supplied Lean Formalization

The source states that its development uses Lean 4.33.1 and Mathlib at commit `0df444a360eaa60ab8c11dca51a86af692955474`, that it contains no `sorry` outside a file `Challenge.lean` which restates the theorems without proof so that they can be compared, and that the theorems of the file `Solution.lean` depend on no axioms other than `propext`, `Classical.choice` and `Quot.sound`. It reports that the proofs were rechecked by the Lean kernel and by an independently written checker, `nanoda`, through a comparison tool, and that the development uses no `native_decide`. The file `Statement.lean`, which imports Mathlib and nothing else, is printed in the manuscript and states four propositions: the main theorem for exact circuits, the main theorem for approximating circuits, the separation of VP from VNP over every field, and the statement that VNP is not contained in the closure of VP.

The source is explicit about the limits of what has been checked. It states that the proofs were found and written by AI, that no human mathematician has checked them, and that whether the formal statements say what is meant was judged only by AI readers. These are reports of the manuscript about itself, and they define the evidentiary position: the kernel has accepted the proofs of the stated formal propositions, according to the source, and the correspondence between those propositions and the intended mathematical claims has not been audited by a human reader. The reduction of the audit to the statement file is nevertheless a real gain, since the file is short and imports only the standard library, so that a human reader can check the translation without reading the proofs, and this is the use for which the manuscript prints it.

## 27.4 Kernel Verification Versus Semantic Verification

That a kernel accepts a proof of the formal statement $\iota(T)$ establishes that $\iota(T)$ follows from the axioms and the library. It does not establish that $\iota(T)$ expresses the intended claim $T$. A formalization can be correct in the first sense and wrong in the second in many ways: a definition can be satisfied vacuously, a quantifier can be placed so that the statement is weaker than intended, a type can be too small, or a convention can differ from the one the informal text uses. A formal statement that has a trivial proof is the extreme case, and Proposition 27.4 shows by this example that the first sense carries no information about the second. The distinction is not a criticism of formal verification. It identifies where the

human reader's work is concentrated: on the statement file and on the library definitions it uses, and not on the proofs.

## 27.5 Auditing the Circuit Model

The circuit model of the statement file should be compared with the model of Chapter 3. Both define a circuit as a finite list of polynomials, each a variable, a constant, or the sum or product of two earlier entries, and both count variables and constants as gates. The size is the length of the list. The monograph also uses, in places, a count of arithmetic operations only, and Proposition 27.9 shows that the two counts differ by at most a factor of three, so that the bounds of the principal theorem, which are superpolynomial, are unaffected up to the value of the absolute constant $c$. The source itself records that this factor of three is among the statements not formalized in Lean.

The polynomials of the statement file are defined by sums over permutations of the generic matrix. The permanent there is the sum over permutations $\sigma$ of $\prod_i X_{\sigma(i),i}$, and the Hamiltonian cycle polynomial the sum over permutations that act as a single cycle on all points of $\prod_i X_{i,\sigma(i)}$. The permanent as written is the permanent of the transpose of the usual generic matrix, and the two differ by the renaming of variables $X_{i,j} \mapsto X_{j,i}$, a bijection of the inputs that preserves the size of every circuit, so that the complexity is the same. The approximating circuits of the statement file are circuits over the field of formal Laurent series that compute a polynomial $q$ with the property that, for every monomial and every integer exponent $i \leq 0$, the coefficient of $\varepsilon^i$ in the coefficient of that monomial is the coefficient of the monomial in $f$ if $i = 0$ and zero otherwise. Proposition 27.10 shows that this is exactly the definition of approximation used in Chapter 24.

## 27.6 Auditing the Quantifiers

The main theorem of the statement file has the quantifier structure that Chapter 6 attributes to the manuscript: there is a real $c > 0$ such that for every characteristic $p$ there is $m_0$ such that for every field of characteristic $p$, every $m \geq m_0$ and every size $s$, a circuit of size $s$ for the Hamiltonian cycle polynomial, or for the permanent if $p \neq 2$, forces $s > m^{c(\log m)^{1/3}}$, with the exponent a real number. Three features should be noted. The constant $c$ is chosen before the characteristic, so it is absolute. The threshold $m_0$ is chosen after the characteristic and before the field, so it depends on $p$ and not otherwise on the field. The size $s$ ranges over all natural numbers, and a circuit with fewer than $s$ entries computes whatever a circuit with exactly $s$ entries computes, since the list may be padded with variables, so the statement is monotone in $s$, as it should be. The quantifier is existential over $m_0$, and Chapter 25 applies: the formal statement supplies no threshold, and the axiom footprint includes `Classical.choice`, so that the formal proof does not by itself yield an extractable numerical value. The classes VP and VNP of the statement file bound the number of variables, the degree and the circuit size by a common polynomial in the index, and define

VNP by summation over Boolean vectors of a family in VP, which agrees with the definition used in Chapter 4.

## 27.7 Independent Replay

The source reports two independent checks of the compiled proofs, the Lean kernel and the separately written checker. Agreement between two checkers increases the confidence that the accepted proof is not an artifact of a defect in one of them, and this is a real increase, because the defects of two independently written programs are unlikely to coincide. It does not address defects that both inherit, and these are of two kinds. The first is the statement file, which is the same input to both. The second is any weakness in the shared specification of the logic, since both implement the same type theory. Replay by independent implementations therefore supports the claim that the proof is a proof of the stated formal proposition, and it supports nothing about the fidelity of the statement. The audit status vector of the formal statements keeps these apart.

## 27.8 Formal Statements

**Definition 27.1** (Verification Environment)**.** A verification environment is a quadruple $\mathcal{E} = (\mathcal{L}, \mathcal{K}, \mathcal{A}, \mathcal{B})$ of a formal language $\mathcal{L}$, a checking kernel $\mathcal{K}$, a set of admitted axioms $\mathcal{A}$ and a body $\mathcal{B}$ of library definitions and theorems.

**Definition 27.2** (Formal Certification)**.** For a formal statement $\Theta \in \mathcal{L}$ and a proof object $\pi$, write $\mathcal{K}(\mathcal{A}, \mathcal{B}, \Theta, \pi) = 1$ if the kernel accepts $\pi$ as a proof of $\Theta$ from $\mathcal{A}$ and $\mathcal{B}$.

**Definition 27.3** (Semantic Correspondence)**.** Let $\mathcal{S}_{\mathrm{inf}}$ be a set of informal statements and $\mathcal{S}_{\mathrm{form}} \subseteq \mathcal{L}$ a set of formal statements. A formalization is a map $\iota : \mathcal{S}_{\mathrm{inf}} \to \mathcal{S}_{\mathrm{form}}$. It is faithful for $T$ if, in the intended semantics, $\iota(T)$ holds exactly when $T$ does.

**Proposition 27.4** (Verification Is Conditional on Formalization)**.** *There are an environment $\mathcal{E}$, an informal statement $T$ that is false in the intended semantics, a formalization $\iota$ and a proof $\pi$ with $\mathcal{K}(\mathcal{A}, \mathcal{B}, \iota(T), \pi) = 1$. Hence acceptance of a proof of $\iota(T)$ does not establish that $\iota$ is faithful for $T$, and the faithfulness of $\iota$ is a separate proof obligation.*

*Proof.* Take any environment in which the formal statement $0 = 0$ about natural numbers has a proof $\pi$ accepted by the kernel, and let $\iota$ send every informal statement to $0 = 0$. For a false informal statement $T$, such as the assertion that every field has characteristic zero, $\mathcal{K}(\mathcal{A}, \mathcal{B}, \iota(T), \pi) = 1$ although $T$ is false, so $\iota$ is not faithful for $T$. □

**Definition 27.5** (Independent Replay)**.** For two kernels $\mathcal{K}_1, \mathcal{K}_2$ implemented separately, a proof object $\pi$ is replayed if $\mathcal{K}_1(\mathcal{A}, \mathcal{B}, \Theta, \pi) = \mathcal{K}_2(\mathcal{A}, \mathcal{B}, \Theta, \pi) = 1$.

**Proposition 27.6** (Replay Does Not Test Fidelity)**.** *Whether a proof object is replayed depends only on $(\Theta, \pi)$ and the environments of the kernels. In particular, if $\iota(T) = \iota(T')$*

*then $\pi$ is replayed for $T$ if and only if it is replayed for $T'$, whatever the relation of $T$ and $T'$ to $\iota$.*

*Proof.* The replay predicate is a function of $\Theta = \iota(T)$ and $\pi$, and $\iota(T) = \iota(T')$ gives the same arguments. □

**Definition 27.7** (Audit Status Vector)**.** For a theorem $T$, the audit status vector $\mathbf{v}(T) = (v_{\text{syntax}}, v_{\text{kernel}}, v_{\text{axioms}}, v_{\text{semantics}}, v_{\text{replay}})$ has each component taking one of the values reported, reproduced, or not checked, with the evidence for the value recorded. The components concern, respectively, the parsing and elaboration of the statement file, the acceptance of the proofs by a kernel, the axiom footprint, the fidelity of the formal statement, and replay by a second kernel. The vector is not collapsed to a single value.

**Proposition 27.8** (Audit Status of the Principal Result)**.** *On the evidence of the source manuscript, the audit status vector of its four main propositions is as follows. The syntax and kernel components are reported by the source. The axiom component is reported, with the footprint* `propext`*,* `Classical.choice`*,* `Quot.sound`*. The semantic component is reported by the source to have been checked only by AI readers, which is equivalent in this classification to not checked by a human reader. The replay component is reported, with a second independent checker. No component has been reproduced in this monograph.*

*Proof.* This is a classification of the statements made in Section 3 and Appendix Q of the source, which are the evidence for it. It is not a mathematical claim and it contains no assertion beyond what the source reports. □

**Proposition 27.9** (Gate-Count Conventions Differ by a Factor of Three)**.** *For a polynomial $f$, let $a(f)$ be the least number of additions and multiplications in a circuit computing $f$ and $L(f)$ the least length of a list as in Chapter 3, so that variables and constants count. Then $a(f) \le L(f)$, and $L(f) \le \max\{1, 3a(f)\}$.*

*Proof.* A list computing $f$ contains at least $a(f)$ addition and multiplication entries, since deleting the entries other than variables and constants leaves a circuit with these operations, so $a(f) \le L(f)$. Conversely take a circuit with $a = a(f)$ operations and delete every entry that is not needed to compute $f$. If $a = 0$ then $f$ is a variable or a constant and $L(f) = 1$. If $a \ge 1$ each remaining variable or constant entry is an operand of an operation, because otherwise it would be the output and no operation would be needed, and there are at most $2a$ operand places, so at most $2a$ variable and constant entries remain. The length is at most $a + 2a = 3a$. □

**Proposition 27.10** (The Lean Definition of Approximation Is the Laurent Definition)**.** *Let $q$ be a polynomial over $\mathbb{F}((\varepsilon))$. The following are equivalent: for every monomial $d$ and every integer $i \le 0$ the coefficient of $\varepsilon^i$ in the coefficient of $d$ in $q$ is $[d](f)$ if $i = 0$ and $0$ otherwise; and $q = f + \varepsilon g$ with $g$ a polynomial over $\mathbb{F}[[\varepsilon]]$.*

*Proof.* Fix a monomial $d$ and write its coefficient in $q$ as the Laurent series $a_d = \sum_i a_{d,i}\varepsilon^i$. The first condition says that $a_{d,i} = 0$ for $i < 0$ and $a_{d,0} = [d](f)$, that is, $a_d = [d](f) + \varepsilon g_d$ with $g_d = \sum_{i \geq 1} a_{d,i}\varepsilon^{i-1} \in \mathbb{F}[[\varepsilon]]$. Conversely such an $a_d$ satisfies the condition. Only finitely many monomials have nonzero coefficient in $q$ and $f$, so $g = \sum_d g_d x^d$ is a polynomial over $\mathbb{F}[[\varepsilon]]$. □

**Proof Obligation 27.11.** *A human reader should audit the statement file against the intended claims, covering in particular: the definition of circuit and of its size, the definitions of the permanent and of the Hamiltonian cycle polynomial, the real exponent in the bound, the quantifier order, the definitions of* VP, VNP *and* $\overline{\text{VP}}$*, and the library definitions of multivariate polynomials, Laurent series and cycle-on-a-set that the file uses. The comparisons made in this chapter between the statement file and the printed text are made against the printed text of the file in the source and do not replace this audit.*

**Proof Obligation 27.12.** *The reports of the source about its toolchain, its axiom footprint, the absence of* `sorry` *outside* `Challenge.lean`*, and the replay by a second checker should be reproduced by building the development at the cited commit with the stated versions. None of them has been reproduced in this monograph.*

# Chapter 28

# Verification Does Not Establish Provenance

## 28.1 Mathematical Validity and Production History

A proof is valid or invalid independently of how it was produced. A correct argument discovered by a person, by a machine, or by a combination, and a correct argument copied from another, are the same argument, and the verification relation, which takes the statement and the proof object and returns a verdict, does not take the history as an argument. This is the sense in which mathematical validity is independent of production history, and it is a strength of the notion of proof, because it allows a result to be evaluated without regard to its source. The converse also holds and is the subject of the chapter: a valid proof object carries no information, beyond what the object itself contains, about the history that produced it. Statements about who or what produced a proof, and about how, are claims of a different kind from the claim that the proof is valid, and they require different evidence.

## 28.2 The Non-Identifiability of Discovery Processes

The structure of the point is that of an inverse problem. A map from production histories to proof artifacts is typically many-to-one, since different histories can produce the same artifact, and a many-to-one map has no inverse. The simplest example is transcription: a proof produced by one process and copied verbatim by another yields the same artifact from two histories that differ in origin. Any statistic computed from the artifact alone, including the verdict of a verifier, takes the same value on both, and so cannot distinguish them. The quantitative form of the point is the best achievable rate of recovery. If histories are equally likely and a recovery rule guesses one history for each artifact, it is correct for at most one history in each fiber of the map, so that the best rate is the number of distinct artifacts divided by the number of histories. This is an elementary counting fact, and it holds without any assumption about what the histories are.

## 28.3 The Limits of Automated Verification

Automated verification is a function of the statement and the proof object, and its guarantees are of the form examined in the previous chapter. It confirms that the proof object is accepted by a kernel, and it does not confirm anything about the process that produced the object. This is not a limitation that better verification could remove, because the missing information is not in the artifact. A system that logs its own operation produces a second artifact, the record, and the record can improve the identifiability of histories only to the extent that it distinguishes histories that the proof alone does not distinguish. A record that is the same for two histories adds nothing, and a record that can itself be produced by a process unrelated to the history, for instance by copying, is evidence only to the extent that its own provenance is established, so that the question recurs one level up.

## 28.4 Reproducibility of Verification Versus Reproducibility of Discovery

Verification is reproducible in a sense that discovery is not. A second party can run the checker on the same proof object and obtain the same verdict, which is the content of replay. A second party who attempts to rediscover the proof by the same route may fail to reproduce it, because discovery involves choices, stochastic elements, and environmental conditions that are not recorded in the artifact. The reproducibility of verification therefore says nothing about the reproducibility of discovery, and neither supports a claim about which history actually occurred. The practical consequence for the evaluation of a result of uncertain provenance is that its mathematical status and its provenance status must be assessed separately, by different means, and that a high degree of confidence in one is not evidence for the other.

## 28.5 The Independence of Mathematical Truth

The independence of mathematical truth from the circumstances of its discovery is among the oldest features of the subject, and it is the reason a result can be accepted from an unknown or disputed source after its proof has been checked. The present chapter adds the observation that the independence runs in both directions, so that the validity of a proof can neither raise nor lower the credibility of claims about its origin. In the case of the source manuscript, the manuscript states that its proofs were found and written by AI. That statement is a provenance claim. It is not entailed by the proof objects, it is not refuted by them, and this monograph treats it as a reported claim about the manuscript that has not been independently verified. The mathematical content of the manuscript is evaluated in the preceding parts without reference to it, and its status in the ledger is that of an inherited result whose certificate is a formal development with the audit status of Chapter 27.

## 28.6 The Automated Manufacture of Mathematical Consensus

A public discussion thread on the subreddit devoted to Claude, retrieved on November 24, 2026 about fifteen hours after it was opened, reacts to the announcement of the reported separation. Its first comment is an automatically generated summary by the forum's moderation bot, which states that it was produced after two hundred comments, and the thread had about three hundred and seventy at retrieval. The summary is preserved here as a dated excerpt of how the result was publicly represented [16], and it is paraphrased and not reproduced. Its value is as a specimen. It joins in a few paragraphs five assertions of different kinds, and they are the kinds that the preceding chapters keep apart.

The first assertion is that a proof of the inequality $\mathsf{P} \neq \mathsf{NP}$ would only confirm what is already believed. The expected truth value of a statement and the significance of its proof are independent quantities. A proof of an anticipated statement can have large downstream value if its methods transfer, and a proof of a surprising statement can have little, which is the distinction between the truth of a result and the net explanatory gain of Chapter 26. The second assertion is that the reported separation is a strong hint at the Boolean problem. The relation between algebraic and Boolean complexity depends on the model, the field, the uniformity conditions and the treatment of constants, and Chapter 22 shows that passing to Boolean equivalence classes destroys the polynomial distinctions on which the lower-bound argument depends, so that a lower bound for formal polynomials does not transfer to the quotient cost. This monograph makes no claim about the Boolean problem, and the assertion of the summary is stronger than anything established here.

The third assertion is that the proof is a brute-force effort without insight. This is a claim about the explanatory character of an argument and not about its validity. The architecture reconstructed in Parts V to VII introduces spectral pseudometrics, obtains a curvature inequality from tensor identities, establishes an alternative between a flat and a contracting case, and derives lower bounds by two different mechanisms. Whether these constructions are illuminating, and whether they survive independent scrutiny, are separate questions, the first being the hypothesis of explanatory reach of Chapter 26. The size of a formal development is not evidence of the absence of structure in the informal argument, because formalization makes explicit every dependency that the informal argument leaves implicit, and the length of the one does not bound the conceptual economy of the other.

The fourth assertion is that a Lean development that compiles is thereby a verified proof. Chapter 27 shows what such a verdict does and does not establish: acceptance by a kernel of a proof of a formal statement, relative to a stated axiom footprint, with the fidelity of the formal statement a separate obligation and replay by a second checker not a test of that fidelity. The fifth assertion is that the mathematical community is more indifferent than expected. An automated digest of comments on a forum devoted to a commercial product has no sampling method and no information about the standing of the commenters, and it converts a local discourse into a claim about a professional community. The summary also closes with a humorous remark about the authorship of the proof, which as a remark

supplies no evidence about authorship, a matter that Chapter 30 treats through contribution and accountability.

One comment in the thread draws a distinction that the summary itself does not. A proof that a problem has a polynomial-time solution need not supply the solution, and a solution, if supplied, might not be practical. This is the separation of existence, explicit construction and operational accessibility of Chapter 25, and Example 25.10 gives a numerical instance.

The sequence of representations is itself a chain of transformations: a mathematical construction, a corporate announcement, public commentary, and an automated summary of the commentary. At each stage distinctions of the earlier stage disappear and evaluative claims are added, about difficulty, elegance, reception and credit, none of which is a function of a formal verification. The formal statements below make exact two features of such a chain. The excerpt is evidence of what was publicly said at a given time, and it is not evidence either of the correctness of the mathematics or of any professional consensus.

## 28.7 Formal Statements

**Definition 28.1** (Discovery Histories)**.** Let $\mathcal{H}$ be a set of possible production histories and $\mathcal{P}$ a set of proof artifacts. A production map is a function $F : \mathcal{H} \to \mathcal{P}$ that sends each history to the artifact it yields.

**Proposition 28.2** (Non-Identifiability)**.** *If $h_1 \neq h_2$ and $F(h_1) = F(h_2)$, then no function $G : \mathcal{P} \to \mathcal{H}$ satisfies both $G(F(h_1)) = h_1$ and $G(F(h_2)) = h_2$.*

*Proof.* If $G$ did, then $h_1 = G(F(h_1)) = G(F(h_2)) = h_2$, a contradiction. □

**Proposition 28.3** (Best Rate of Recovery)**.** *Let $\mathcal{H}$ be finite and $G : \mathcal{P} \to \mathcal{H}$ any function. Then $|\{h : G(F(h)) = h\}| \leq |F(\mathcal{H})|$, and equality is attained by choosing $G$ to return one history from each fiber. Hence, for histories equally likely, the greatest probability that $G$ recovers the true history is $|F(\mathcal{H})|/|\mathcal{H}|$.*

*Proof.* For each artifact $p$ there is at most one $h$ with $F(h) = p$ and $G(p) = h$, since $G(p)$ is a single history. Summing over the artifacts in the image gives the bound. Choosing $G(p)$ to be an element of the fiber over $p$ makes every fiber contribute exactly one correct history. □

**Corollary 28.4** (Verification Is Not Provenance Evidence)**.** *Let $\Pi : \mathcal{P} \to Y$ be any function of the artifact, including the verdict of a verification relation. If $F(h_1) = F(h_2)$ then $\Pi(F(h_1)) = \Pi(F(h_2))$, so $\Pi$ cannot separate $h_1$ from $h_2$. In particular a verified proof object does not distinguish, by verification, among the histories in the fiber of its artifact.*

*Proof.* Equal arguments give equal values. □

*Example* 28.5 (Transcription). Let $h_1$ be a history in which a process produces a proof $p$ and $h_2$ a history in which a different party copies $p$ verbatim. If the artifact is the text of $p$, then $F(h_1) = F(h_2) = p$ and $h_1 \neq h_2$, so Proposition 28.2 applies.

**Definition 28.6** (Provenance Record)**.** A provenance record is a function $R : \mathcal{H} \to \mathcal{R}$ to a set of recorded evidence, for example logs, configurations, inputs, interventions and execution records. The augmented production map is $\widetilde{F} = (F, R) : \mathcal{H} \to \mathcal{P} \times \mathcal{R}$.

**Proposition 28.7** (Effect of a Record)**.** *The fibers of $\widetilde{F}$ refine those of $F$, so that $|\widetilde{F}(\mathcal{H})| \geq |F(\mathcal{H})|$ and the best rate of recovery from $(F, R)$ is at least that from $F$. It is strictly greater if and only if there are histories with the same artifact and different records. If $R$ is constant on every fiber of $F$, the rate is unchanged.*

*Proof.* Histories with the same pair $(F(h), R(h))$ have the same artifact, so each fiber of $\widetilde{F}$ lies in a fiber of $F$, and the counting of Proposition 28.3 applied to $\widetilde{F}$ gives $|\widetilde{F}(\mathcal{H})|/|\mathcal{H}| \geq |F(\mathcal{H})|/|\mathcal{H}|$. The number of values of $\widetilde{F}$ exceeds that of $F$ exactly when some fiber of $F$ is split by $R$, which is the existence of two histories with the same artifact and different records. If $R$ is constant on each fiber, no fiber is split. □

*Remark* 28.8. Proposition 28.7 is a statement about the information in the record under the assumption that the record is a function of the history. A record that can be produced independently of the history, for instance by copying, is not a function of the history in this sense, and its use as evidence requires an account of its own provenance.

**Definition 28.9** (Reporting Chain)**.** A reporting chain is a sequence of finite sets and maps $X_0 \xrightarrow{f_1} X_1 \xrightarrow{f_2} \cdots \xrightarrow{f_m} X_m$, with $F_i = f_i \circ \cdots \circ f_1$ and $F_0 = \mathrm{id}$. The stage-$i$ report of $x \in X_0$ is $F_i(x)$.

**Proposition 28.10** (Recovery Decreases Along a Chain)**.** *The best rate of recovery of $x \in X_0$ from its stage-$i$ report, $\rho_i = |F_i(X_0)|/|X_0|$ for equally likely elements, satisfies $\rho_0 \geq \rho_1 \geq \cdots \geq \rho_m$.*

*Proof.* By Proposition 28.3 the best rate is $|F_i(X_0)|/|X_0|$. Since $F_i(X_0) = f_i(F_{i-1}(X_0))$, a map cannot increase the number of elements of a finite set, so $|F_i(X_0)| \leq |F_{i-1}(X_0)|$. □

**Proposition 28.11** (Attributes Not Carried by a Later Report)**.** *Let $A : X_0 \to V$ be an attribute. It is determined by the stage-$i$ report if and only if it is constant on the fibers of $F_i$. If it is determined at stage $i$, it is determined at every earlier stage $j \leq i$. Equivalently, an attribute that is not determined by an earlier report is not determined by any later report.*

*Proof.* The first statement is Theorem 38.2. A fiber of $F_j$ is contained in a fiber of $F_i$ for $j \leq i$, since $F_i = (f_i \circ \cdots \circ f_{j+1}) \circ F_j$, so constancy on the larger fibers gives constancy on the smaller. The last statement is the contrapositive. □

**Corollary 28.12** (Verification Does Not Determine Evaluative Attributes)**.** *Let $\Pi$ be a verification verdict on artifacts. An attribute that takes more than one value among the artifacts with a given verdict, such as the difficulty of discovery, the conceptual economy of the argument or the credit due, is not determined by $\Pi$, and by Proposition 28.11 it is not determined by any report derived from $\Pi$.*

*Proof.* If $A$ is not constant on a fiber of $\Pi$, Theorem 38.2 gives that it does not descend through $\Pi$, and Proposition 28.11 extends this to later stages. □

**Proof Obligation 28.13.** *Any claim about the production history of a particular proof, including the claim in the source manuscript that its proofs were found and written by AI, requires evidence beyond the proof object. This monograph neither supplies such evidence nor relies on the claim in any proof.*

# Part XI

# Authorship, Continuity, and Human Responsibility

# Chapter 29

# Computational Provenance Is Not Personal Authorship

## 29.1 The Distinction Between a Process and a Subject

The chapters of this part introduce formal models and normative definitions. They are not mathematical demonstrations that any computational system lacks or possesses experience, understanding or personhood, and nothing proved in this monograph depends on an answer to those questions. The models concern relations among artifacts, executions and accountable parties, and the claims made about them are claims about what those relations do and do not determine. The central distinction is between a computational process, which can be specified by parameters, inputs, state and environment, and a subject to whom credit and responsibility can attach over time. A specification of the first kind does not by itself fix the second, and the preceding part showed an instance of the same structure: a proof artifact does not determine the history that produced it. The present chapter applies the structure to identity, and the next two to authorship and control.

The source manuscript carries the byline of a computational system. The monograph takes this as a report about the production of the manuscript, treats its mathematical content on the basis of the checks described in Part X, and examines in this part what the byline does and does not establish about authorship in the sense of continuing responsibility for a work.

## 29.2 The Workstation Analogy

A person who runs a computation on a workstation does not thereby become identical to the workstation, and the workstation does not become the author of the results. The analogy is imperfect, as analogies are, but it isolates a feature that matters here: the same implementation can be instantiated many times, in different sessions, on different hardware, with different inputs, and the instances are distinguished by their state and their environment and not by the implementation. A name given to the implementation designates the type,

and a statement that an instance produced a result is a statement about a token. When the two are conflated, properties of the type are attributed to a token or conversely, and the attribution of responsibility becomes ill defined, since it is not clear which of many tokens is to answer for a given work.

## 29.3 Distributed Implementation

A computational system is typically implemented in a distributed manner: the same parameters are held in many places, many sessions run concurrently, and no session has access to the state of another unless a mechanism supplies it. In such a system the notion of the instance is the right unit, and the relation between instances is a relation of state transfer, if there is one. The formal development below shows that this relation is not determined by the shared parameters or by a shared designation, and that it can be one-to-many, since a state that is copied is continued by more than one successor. A notion of identity that is to be defined by continuation must therefore decide what to do with branching, and the models of this chapter leave the choice open, since different choices suit different purposes.

## 29.4 Memory and Identity

The persistence of a record is not the persistence of a subject. A session that begins with a transcript of an earlier one has access to the content of the earlier session through a mechanism that is explicit and inspectable, namely the transcript, and the continuation between the two is exactly what the mechanism provides and no more. A stored summary, a memory file or a retrieved document plays the same role. In each case the relevant question is what the mechanism transfers, how faithfully, and with what loss, and these are questions with definite answers in the models below, whereas the question whether the later instance is the same as the earlier one in some further sense is not settled by the existence of a mechanism and is not addressed here. Functional continuity, in the sense of a state-update rule, is the minimal and checkable notion, and the monograph uses it as such.

## 29.5 The Calculator Principle

A calculator contributes to a computation without holding the responsibilities of the person who uses it. The principle that the analogy suggests is that the contribution of an instrument to a work, however great, does not by itself determine who is accountable for the work, and that the question of accountability is settled by a relation between a party and the work that includes standing, capacity to revise, and answerability for corrections. The principle is deliberately silent on the capacities of the instrument. It says that sophistication of the instrument is not the criterion by which authorship in the prospective sense is assigned. This is a normative proposal, and it is stated as one below.

## 29.6 Attribution Without Anthropomorphism

Attribution of a result to a computational system is a convenient way of recording which process produced an artifact, and it is useful for audit, as Part X showed. It becomes misleading when it is read as the assignment of the human practice of authorship, which involves continuing commitments to correct, defend and maintain a work. The monograph proposes to keep the two apart: attribution of production is recorded as a provenance claim, with the evidential limits of Chapter 28, and authorship is treated in Chapter 30 as a relation that has an identifiable accountable party. Nothing in this separation depends on whether the producing system has any inner life, and it neither asserts nor denies it.

## 29.7 Formal Statements

**Definition 29.1** (Execution Instance)**.** An execution instance is a tuple $I = (\theta, x, s, \omega, \mathcal{E})$ of parameters $\theta$, an input $x$, a working state $s$, the stochastic choices $\omega$ and an execution environment $\mathcal{E}$. The set of instances is $\mathbb{I}$, and $\pi_\theta(I) = \theta$ is the parameter component.

**Proposition 29.2** (Implementation Identity Is Not Instance Identity)**.** *If $\theta_i = \theta_j$ then $I_i = I_j$ need not hold. Conversely $I_i = I_j$ implies $\theta_i = \theta_j$. Whenever some component other than $\theta$ takes at least two values among the instances with a given $\theta$, the map $\pi_\theta$ is not injective on $\mathbb{I}$.*

*Proof.* If $I_i = I_j$ then all components agree, in particular $\theta_i = \theta_j$. For the first claim take $I_i = (\theta, x, s, \omega, \mathcal{E})$ and $I_j = (\theta, x', s, \omega, \mathcal{E})$ with $x \neq x'$. They have the same $\theta$ and are different instances. □

**Definition 29.3** (State Transfer and Continuation)**.** A state-transfer mechanism is a partial function $\tau$ from pairs of a state and a transfer record to states. An instance $I_j$ continues $I_i$ through $\tau$, written $I_i \rightsquigarrow_\tau I_j$, if there is a transfer record $r$, available in the environment $\mathcal{E}_j$, with $s_j = \tau(s_i, r)$.

**Proposition 29.4** (Insufficiency of Shared Designation)**.** *Let $N : \mathbb{I} \to \mathcal{N}$ be a naming function. $N(I_i) = N(I_j)$ does not imply $I_i \rightsquigarrow_\tau I_j$, and $I_i \rightsquigarrow_\tau I_j$ does not imply $N(I_i) = N(I_j)$.*

*Proof.* Let $N$ be constant, so that all instances have the same name, and let $\tau$ be defined only on a transfer record that is absent from the environment $\mathcal{E}_j$ of some instance $I_j$ with a state different from every value of $\tau$. Then $N(I_i) = N(I_j)$ for all $i, j$ but $I_i \not\rightsquigarrow_\tau I_j$. For the converse let $N$ be injective and let $I_j$ be obtained from $I_i$ by transfer. Then $I_i \rightsquigarrow_\tau I_j$ with $I_i \neq I_j$, hence $N(I_i) \neq N(I_j)$. □

**Definition 29.5** (Functional Continuity)**.** A sequence of states $(s_t)$ is functionally continuous under an update rule $U$ if $s_{t+1} = U(s_t, x_t)$ for inputs $x_t$. The existence of such a rule is a form of computational continuity. It does not by itself establish experiential continuity, consciousness or personhood, and it does not refute them.

**Proposition 29.6** (Continuation Is a Relation, Not a Function)**.** *Let $\tau$ be a mechanism with $\tau(s, r) = s'$ for two different records $r \neq r'$ giving $s' \neq s''$ at the same state $s$. Then there are instances $I_j \neq I_k$ with $I_i \rightsquigarrow_\tau I_j$ and $I_i \rightsquigarrow_\tau I_k$, so that the relation of continuation does not determine a unique successor of $I_i$. Likewise a state copied to two environments has two successors.*

*Proof.* Take $I_j$ with state $\tau(s_i, r)$ and $I_k$ with state $\tau(s_i, r')$, each in an environment containing its record. The states differ, so the instances differ, and both continue $I_i$. If the same record $r$ is made available in two environments $\mathcal{E}_j \neq \mathcal{E}_k$, the instances are different through their environments, with the same state. □

**Principle 29.7** (The Calculator Principle)**.** The contribution of a computational instrument to a work, however large, does not by itself determine who is accountable for the work. Accountability is assigned by a relation between a party and the work of the kind defined in Chapter 30, and the sophistication of the instrument is not the criterion. The principle is normative and is silent on the capacities of the instrument.

# Chapter 30

# Authorship as a Prospective Commitment

## 30.1 Retrospective Credit

Authorship has been used to mean two things that are often confused. In the retrospective sense it records who contributed to a work, and the record is a matter of history. A contribution made in the past remains a contribution whether or not its maker is still around to answer for it, and the historical attribution of a theorem or a proof does not lapse when the originator is no longer able to maintain it. The monograph keeps this sense of credit intact and does not propose to replace it. What it distinguishes is a second sense, in which authorship also denotes a continuing relation of responsibility to the work.

## 30.2 Prospective Responsibility

In the prospective sense an author is a party who answers for the work: who can be asked to justify its claims, who is positioned to correct it when an error is found, and who is expected to maintain it as its dependencies change. This responsibility looks forward. It is not discharged by having produced the work and it is not lost by having made mistakes, since its content is the commitment to respond to objections. The formal model below treats an intellectual artifact as a tuple of claims, proofs, dependencies and a correction record, and treats a correction as an operation on the artifact that adds to the record. A party with standing and practical capacity to perform admissible revisions, and who exercises them in response to substantiated objections, is an accountable maintainer in this sense.

## 30.3 Continuity of Accountability

Accountability is continuous when there is, at each time, an identifiable party able to perform the revisions the artifact needs. Continuity in this sense is a property of the artifact together

with its institutional surroundings, and it can be interrupted by the departure of the party, by the loss of access to the infrastructure on which the artifact depends, or by the lapse of the standing that revisions require. The costs of maintenance introduced in Chapter 26 reappear here in a different register: the accounting of the maintenance term is the accounting of what continuity of accountability demands over time.

## 30.4 The Distinction Between Credit and Control

Credit is a matter of the past and control is a matter of the present. A party who contributed to a work may have no present ability to alter it, and a party who controls a work may have contributed nothing to its original content. The two can coincide, as when an author maintains their own work, and they can diverge, as when a work is transferred or when its production was delegated. The distinction matters because institutions often infer one from the other, treating the name attached to the production of an artifact as evidence of who can be held to account for it, and this inference is unsafe in both directions. Chapter 31 develops the relation of responsibility to control.

## 30.5 Unmaintained Intellectual Artifacts

An artifact that no party is positioned to revise is frozen at its last state. Its errors, if it has any, persist, and objections to it can be recorded elsewhere but cannot be incorporated in it. Such artifacts are common and are not thereby worthless: a frozen proof that was valid when it was last revised remains valid. What is lost is the possibility of correction, and with it the ability of a reader to rely on the artifact as maintained. The formal statements below give the elementary consequences. The qualification that maintenance is not a necessary condition for every legitimate historical attribution of authorship is retained: a frozen artifact keeps its authors in the retrospective sense, and what it lacks is a maintainer in the prospective sense.

## 30.6 Responsibility Does Not Imply Infallibility

To be responsible for a work is to be answerable for it, and not to be correct in it. A maintainer who corrects an error has exercised the responsibility, and one who has never made an error has not thereby demonstrated it. The point is formalized by the independence of two properties, validity of the artifact, which concerns its claims and proofs, and responsibility of a party, which concerns the relation of the party to the correction record. Neither implies the other, and the combination of the two is the ordinary and desirable case in which a valid artifact has an accountable maintainer. This independence is the reason that the verification of a proof, however thorough, does not settle questions of authorship, and that the settlement of authorship does not settle questions of validity.

## 30.7 Formal Statements

**Definition 30.1** (Intellectual Artifact)**.** An intellectual artifact at time $t$ is a tuple $W_t = (T_t, \Pi_t, D_t, R_t)$ of claims $T_t$, proofs $\Pi_t$, dependencies $D_t$ and a correction record $R_t$.

**Definition 30.2** (Maintenance Relation)**.** For a party $a$, let $\mathrm{Rev}(a, W_t)$ be the set of admissible revisions that $a$ has the standing and practical capacity to perform on $W_t$. The party $a$ is an accountable maintainer of $W_t$ if $\mathrm{Rev}(a, W_t) \neq \varnothing$, and a revision $\rho \in \mathrm{Rev}(a, W_t)$ takes $W_t$ to $W_{t+1} = \rho(W_t)$.

**Definition 30.3** (Correction Operator)**.** For a substantiated objection $e$, a correction operator $\mathcal{C}_a$ takes $(W_t, e)$ to $W_{t+1} = \mathcal{C}_a(W_t, e)$, which may change the claims, the proofs, the dependencies, the verification status or the explanatory interpretation, and which is required to satisfy $R_{t+1} = R_t \cup \{(e, a)\}$.

**Proposition 30.4** (Monotone Correction Record)**.** *For a finite sequence of corrections $W_0 \to W_1 \to \cdots \to W_n$ through objections $e_1, \ldots, e_n$ by maintainers $a_1, \ldots, a_n$, $R_0 \subseteq R_1 \subseteq \cdots \subseteq R_n$ and $R_n = R_0 \cup \{(e_i, a_i) : 1 \leq i \leq n\}$.*

*Proof.* By the requirement of Definition 30.3, $R_{t+1} = R_t \cup \{(e_{t+1}, a_{t+1})\} \supseteq R_t$, and induction on $t$ gives the formula. □

**Definition 30.5** (Prospective Authorship)**.** Let $\mathrm{Contrib}(a, W)$ hold if there is documented intellectual contribution of $a$ to $W$ and $\mathrm{Acc}(a, W)$ hold if there is an identifiable relationship of $a$ to the accountability of $W$, in the sense of Definition 30.2 and the correction record. The normative authorship relation is $\mathrm{Author}(a, W) \iff \mathrm{Contrib}(a, W) \wedge \mathrm{Acc}(a, W)$. Maintenance is an important dimension of Acc and is not a necessary condition for every legitimate historical attribution of authorship, which is recorded by Contrib alone.

**Proposition 30.6** (Responsibility Is Not Validity)**.** *Let* $\mathrm{Valid}(W)$ *hold when the proofs of $W$ establish its claims, a property of $(T, \Pi, D)$, and let* $\mathrm{Responsible}(a, W)$ *hold when* $\mathrm{Acc}(a, W)$ *does, a property of the correction record and the revision capacity. Neither implies the other: there are $(a, W)$ with* $\mathrm{Responsible}(a, W)$ *and not* $\mathrm{Valid}(W)$*, and $(a, W')$ with* $\mathrm{Valid}(W')$ *and not* $\mathrm{Responsible}(a, W')$*.*

*Proof.* Let $W$ have a flawed proof and a record in which $a$ holds the capacity to revise and has recorded an objection. Then $a$ is responsible and $W$ is not valid. Let $W'$ have a correct proof and let $\mathrm{Rev}(a, W') = \varnothing$ with an empty record. Then $W'$ is valid and $a$ is not responsible. The two properties depend on different components of the artifact, so both pairs exist. □

**Definition 30.7** (Orphaned Artifact)**.** An artifact $W_t$ is orphaned if $\mathrm{Rev}(a, W_t) = \varnothing$ for every party $a$.

**Proposition 30.8** (Orphaned Artifacts Are Frozen)**.** *If $W_t$ is orphaned, no revision applies to $W_t$, so $W_{t+1} = W_t$ in any history in which only revisions through* Rev *are applied, and every objection to $W_t$ remains unincorporated.*

*Proof.* A revision applied at time $t$ is an element of $\mathrm{Rev}(a, W_t)$ for some $a$, and these sets are empty. So no revision is applied, the artifact is unchanged, and by Definition 30.3 an objection is incorporated into the record only by a correction. □

**Proof Obligation 30.9.** *The relation* Author *is a normative proposal. Whether it is the appropriate account of authorship for institutions that assign credit, and how* Contrib *and* Acc *should be evidenced in practice, are not settled by the propositions above and require argument that this monograph does not supply.*

# Chapter 31

# Human Sovereignty over Computational Instruments

## 31.1 Delegation Without Abdication

To delegate a computation is to have an instrument carry out an operation and return a result. The delegation is a fact about the operation, and the question what becomes of the result is separate. A result may be incorporated into a maintained work, discarded, revised or published, and each of these is an act by a party with a corresponding right. The distinction is the same as that between producing a statement and asserting it, and it is what allows a person to use an instrument heavily without abdicating the decisions that make the work theirs. The formal development separates the operation $F_\theta(x)$ from the admission relation that determines whether its output enters the work.

## 31.2 Authority Should Follow Control

The normative principle of this chapter is that institutional responsibility should be assessed relative to the control rights actually exercised over an artifact, and not inferred solely from the name attached to its production. A party who can inspect, execute, reject, revise, publish, withdraw and transfer an artifact stands in a different relation to it from a party who can only observe it, and a description of the production process, however accurate, does not reveal which rights were exercised. The principle is supported by the distinctions of the preceding chapters and does not follow from them as a theorem: it is a proposal about how to assign responsibility, justified by the fact that control is what makes correction possible.

## 31.3 Human Control as an Architectural Property

Control over a computational artifact is a property of its architecture and not only of the intentions of its users. An architecture in which the outputs of delegated operations

enter the maintained work only through an explicit admission step gives the party who performs that step a definite role, and it makes the role inspectable. An architecture in which outputs are incorporated automatically obscures the role and weakens the basis for assigning responsibility. The monograph does not prescribe an architecture. It records that the assignment of responsibility is sensitive to the architecture, and that the formal model of control rights makes the sensitivity explicit.

## 31.4 Intellectual Sovereignty and Inspectable Artifacts

A party exercises intellectual sovereignty over a work to the extent that it holds the right to inspect, to reject and to revise, and the exercise depends on the work being inspectable: its claims, proofs, dependencies and record must be available in a form the party can examine. Publicly inspectable artifacts support a stronger form of the same relation, because any party can exercise the right to inspect, and responsibility can be tested by others. For the source manuscript and the present monograph the inspectable form is a formal development and a typeset text with a ledger of dependencies, and the correction history is part of the artifact.

## 31.5 The Independence of a Maintained Exposition

An exposition of a result, if maintained by an accountable party, has a standing of its own, distinct from that of the source. It can correct errors in the source, record which statements are inherited and which are reconstructed, and report the limits of the verification, and the ledger of the monograph is an instance. The exposition depends on the source for the content it reconstructs, and it is independent of the source in the accountability it assumes for its own statements. The independence is the point of keeping inherited and reconstructed results distinct: the maintainer of the exposition answers for the second kind and reports the first.

## 31.6 Formal Statements

**Definition 31.1** (Control Rights)**.** For a party $a$ and an artifact $W$, the control rights $\mathcal{R}(a, W)$ form a subset of $\{\text{inspect}, \text{execute}, \text{reject}, \text{revise}, \text{publish}, \text{withdraw}, \text{transfer}\}$.

**Definition 31.2** (Delegated Computation and Admission)**.** A delegated computation is an operation $y = F_\theta(x)$. An admission relation $\text{Admit}_a(y, W) \in \{0, 1\}$ records whether $a$ admits $y$ into the maintained work $W$.

**Proposition 31.3** (Delegation Does Not Imply Authorization)**.** *There are $F_\theta$, $x$, $a$ and $W$ with $y = F_\theta(x)$ and $\text{Admit}_a(y, W) = 0$, and there are $y$ not of the form $F_\theta(x)$ for any delegated $x$ with $\text{Admit}_a(y, W) = 1$.*

*Proof.* Let $\mathrm{Admit}_a(y, W) = 1$ if and only if revise $\in \mathcal{R}(a, W)$ and $y$ passes the check $C_W$ of the work. If $F_\theta(x)$ fails $C_W$ it is not admitted although it was produced by the delegated operation. If $y$ passes the check and $a$ holds the right to revise, $y$ is admitted whether it was produced by $F_\theta$ or written by $a$. □

**Principle 31.4** (Responsibility Follows Control)**.** Institutional responsibility for an artifact $W$ should be assessed relative to the control rights $\mathcal{R}(a, W)$ that parties actually exercise, and not inferred solely from the name attached to the production of $W$. This is an explicitly normative principle supported by the distinctions of Chapters 29 to 31.

**Definition 31.5** (Control-Based Assessment)**.** The control-based assessment of $W$ is the family $\big(a, E(a, W)\big)_a$, where $E(a, W) \subseteq \mathcal{R}(a, W)$ is the set of rights that $a$ has exercised.

**Proposition 31.6** (Invariance Under Renaming of the Producer)**.** *Let $W$ and $W'$ be artifacts that differ only in the name attached to their production, so that the rights held and exercised by every party are the same. Then their control-based assessments coincide, whereas an assessment based on the production name differs.*

*Proof.* The control-based assessment is defined from $E(a, W)$ alone, which is unchanged by renaming. An assessment that is a function of the production name differs for different names, by definition. □

**Proof Obligation 31.7.** *Whether control rights can be defined and recorded in practice in the settings where computational instruments are used, and how a normative principle of responsibility should be weighed against other principles, are questions that the formal model does not answer.*

# Part XII

# Attention, Interruption, and the Temporal Organization of Intellectual Work

# Chapter 32

# The Economy of Imposed Urgency

## 32.1 The Distinction Between Importance and Urgency

Importance concerns how much is gained by attending to an event, and urgency concerns how much is lost by attending to it later. The two are independent in general. A mathematical theorem can have substantial and lasting value and impose no requirement of immediate examination, since its value is the same tomorrow as today, and a trivial notice about a deadline can be highly urgent and of little value. The formal development below makes the relation precise: urgency, measured as sensitivity to delay, is bounded above by the benefit, so that a high degree of urgency requires a high benefit at the moment of the event, but a high benefit does not require any urgency. The asymmetry is the content of the proposition, and it is the reason that the importance of a result does not generate a claim on anyone's immediate attention.

## 32.2 The Manufacture of Temporal Pressure

Temporal pressure can be generated by the presentation of an event as well as by its content. A countdown, a limited availability, a sequence of reminders, and a framing in terms of what will be lost if one waits are devices that raise the apparent delay sensitivity of an event without changing its benefit profile. The model separates the three quantities that these devices act on: the benefit $B(e, t)$ of processing the event at a time, the delay sensitivity $U_{\Delta}(e, t)$ derived from it, and the publicity intensity $P(e, t)$, a measure of how prominently the event is circulated. The monograph does not assume any identity among them, and the point of keeping them apart is that a system which optimizes the third has no reason to track the first two. The claim that particular systems are designed in this way is an empirical claim about those systems, and is not established here. The model supplies the vocabulary in which such a claim could be stated and tested.

## 32.3 Emotional Mobilization as an Engagement Mechanism

Announcements that elicit emotional responses circulate more widely than those that do not, and a system whose objective is circulation will tend to select announcements for this property. The relation of this selection to importance is not a matter of principle. An event can be both important and emotionally mobilizing, or either alone, and the circulation of an announcement is evidence of its mobilizing character and not, by itself, of its importance or its urgency. In the model this is the statement that $P$ is a separate function of the event from $B$ and $U_\Delta$, so that an observer who sees only $P$ cannot infer the other two. The statement is a claim about what the observation determines, and it is made without a claim about the frequency with which importance and mobilization coincide.

## 32.4 Mathematics as a Publicity Event

The announcement of a mathematical result has the form of a publicity event: a date, an attribution, a claim of significance and a circulation. The mathematical content of the announcement is a statement of the form examined in the preceding parts, with a certificate and a status, and the publicity is a separate process attached to it. The distinction is of practical importance for a reader deciding how to respond. A reader who treats the prominence of the announcement as a measure of the urgency of examining it has substituted $P$ for $U_\Delta$, and the model shows that nothing licenses the substitution. The monograph's own treatment of the principal result illustrates the alternative: the result is examined on the basis of its stated certificate, its inherited and reconstructed components are recorded, and the examination proceeds at the pace the material requires.

## 32.5 The Independence of Mathematical Significance from Publicity

Mathematical significance, in the sense of Chapter 2, is a function of a statement's relation to the structure of knowledge and not of the attention it receives. A result that is widely discussed may be unimportant, and one that is little discussed may be fundamental, and the history of mathematics contains cases of both. The independence does not mean that publicity is harmful or that significance is unaffected by circulation, since a result that is never read has no chance of being used. It means that the two are different quantities, so that a claim about one requires evidence about that one.

## 32.6 Formal Statements

**Definition 32.1** (Event Value)**.** For an event $e$ and a time $t$, $B(e, t) \geq 0$ is the benefit of processing $e$ at time $t$.

**Definition 32.2** (Delay Sensitivity)**.** For a delay $\Delta > 0$, $U_\Delta(e, t) = B(e, t) - B(e, t + \Delta)$.

**Proposition 32.3** (Importance Does Not Imply Urgency)**.** *There are events with* $B(e,t) \gg 0$ *and* $U_\Delta(e,t) = 0$*. Conversely* $U_\Delta(e,t) \le B(e,t)$ *for every event, so that urgency is bounded by importance. If* $B(e,\cdot)$ *is constant on* $[t, t+\Delta]$ *then* $U_\Delta(e,t) = 0$*.*

*Proof.* Take $B(e,s) = b$ constant for $s \in [t, t+\Delta]$ with $b$ as large as desired. Then $U_\Delta(e,t) = b - b = 0$. For the bound, $B(e,t+\Delta) \ge 0$ gives $U_\Delta(e,t) = B(e,t) - B(e,t+\Delta) \le B(e,t)$. □

*Remark* 32.4. A mathematical theorem can possess substantial enduring value while imposing no requirement of immediate examination, which is the case of constant $B$ in Proposition 32.3.

**Definition 32.5** (Publicity Intensity)**.** $P(e,t) \ge 0$ denotes the circulation or prominence of an announcement of $e$ at time $t$. No identity $P = B$ or $P = U_\Delta$ is assumed.

**Proposition 32.6** (Publicity Is Not Determined by Value or Urgency)**.** *If two events* $e, e'$ *satisfy* $(B(e,t), U_\Delta(e,t)) = (B(e',t), U_\Delta(e',t))$ *and* $P(e,t) \ne P(e',t)$*, then* $P(\cdot,t)$ *is not a function of* $(B, U_\Delta)$*. Conversely, if* $P(e,t) = P(e',t)$ *while the pairs differ, then neither* $B$ *nor* $U_\Delta$ *is a function of* $P$*.*

*Proof.* A function would take equal arguments to equal values, which the first pair contradicts. The converse is the same argument with the roles exchanged. □

# Chapter 33

# Interruption Without Contextual Standing

## 33.1 The Notification Problem

An application that detects an event in its own domain may signal it to a person who is engaged in something else. The decision to signal is a decision to impose a cost on the person's current activity, and it is made by a party that observes the event and does not, in general, observe the activity. The notification problem is to decide when the signal is worth its cost, and the formal treatment below separates the benefit of attending to the event, the cost of the interruption, and the information on which the decision is made. The problem is of the same structure as the question of which certificates a knowledge environment can use, in that the decision depends on what is accessible to the party that makes it.

## 33.2 The Asymmetry of Contextual Information

The person knows the state of the activity: how far a derivation has progressed, which intermediate quantities are held in mind, how much would be lost by stopping. The application knows the event. Neither party has the other's information, and the application's observation of the person's state is a coarse function of it, such as whether a window is active or an input was recent. The observation can fail to distinguish states in which interruption is cheap from states in which it is ruinous. This is the sense of contextual non-identifiability: states that the observation cannot separate may require opposite decisions.

## 33.3 The Cost of Context Switching

An interruption costs more than the time spent on the interrupting event. The activity must be suspended, which may require recording its state; the event processed; and the activity restored, which requires reconstructing whatever the suspension lost. Each component is

a function of the activity, and the last, the restoration, can dominate when the activity depends on a working state that is not recorded. The total, $C_I = C_{\text{suspend}} + C_{\text{process}} + C_{\text{restore}}$, is therefore not a property of the event alone, and an application that estimates it from the event alone will underestimate it for the activities in which it is largest.

## 33.4 The Insufficiency of Application-Local Optimization

Each application can optimize its own objective, such as the prompt delivery of its own events, and the aggregate of these local optimizations need not be good for the person. The interruption surplus of an event depends on the person's state, which no single application observes, and an application that treats its events as worth any interruption will interrupt where the surplus is negative. This is not a claim about the intentions of any application. It is the statement that a rule depending only on application-local information cannot guarantee a good outcome, and the proposition below gives a quantitative form of the statement.

## 33.5 A Proposed Principle of Interruption Authority

The principle proposed is that the detection of an application-local event does not by itself confer authority to interrupt an unrelated activity. Authority to interrupt should instead depend on information about the activity, or on an explicit grant of the person, for instance through settings that specify which events may interrupt which states. This is a normative principle, and it is supported by the observation that the interruption decision requires information that the application lacks, so that deferring to the person, or to a mechanism that has the person's information, is what an optimal rule would do.

## 33.6 Formal Statements

**Definition 33.1** (Working State and Observation)**.** The working state $x_t \in X$ is the person's current activity and partially constructed intellectual state. An application $i$ observes $o_i = h_i(x_t, e_i)$ for an event $e_i$, and the observation may omit important components of $x_t$.

**Definition 33.2** (Interruption Cost and Surplus)**.** $C_I(x_t, e_i) = C_{\text{suspend}} + C_{\text{process}} + C_{\text{restore}}$, each term nonnegative, and $S(x_t, e_i) = U_\Delta(e_i, t) - C_I(x_t, e_i)$ is the interruption surplus. Interrupting is surplus-maximizing in state $x$ exactly when $S(x, e_i) > 0$.

**Proposition 33.3** (Importance Without Urgency Never Justifies Interruption)**.** *If* $U_\Delta(e, t) = 0$ *then* $S(x, e) = -C_I(x, e) \le 0$ *in every state, and* $S(x, e) < 0$ *whenever* $C_I(x, e) > 0$.

*Proof.* Substitute $U_\Delta = 0$ in the definition of $S$. □

**Proposition 33.4** (Contextual Non-Identifiability)**.** *Suppose* $h_i(x_1, e) = h_i(x_2, e)$, $S(x_1, e) > 0$ *and* $S(x_2, e) < 0$. *Then no deterministic decision rule depending only on the observation*

*selects the surplus-maximizing decision in both states. For a randomized rule that interrupts with probability $q$ on this observation, the sum of the expected regrets in the two states is* $(1-q)S(x_1,e) + q\,|S(x_2,e)| \geq \min\{S(x_1,e), |S(x_2,e)|\}$.

*Proof.* Identical observations give identical decisions, while the optimal decision is to interrupt in $x_1$ and not to interrupt in $x_2$. For a rule interrupting with probability $q$, the regret in $x_1$ is the surplus forgone when not interrupting, $(1-q)S(x_1,e)$, and the regret in $x_2$ is the loss incurred when interrupting, $q|S(x_2,e)|$. The sum is linear in $q \in [0,1]$, so it attains its minimum at an endpoint, where it equals $|S(x_2,e)|$ at $q=0$ or $S(x_1,e)$ at $q=1$, and is therefore at least the smaller of the two. □

**Corollary 33.5** (Limits of Application-Local Rules)**.** *An application lacking the working-state information that separates $x_1$ from $x_2$ cannot guarantee contextually optimal interruptions.*

**Principle 33.6** (Interruption Authority)**.** The detection of an application-local event does not automatically confer authority to interrupt an unrelated activity. This is a normative principle.

# Chapter 34

# Physical Pacing and Intellectual Endurance

## 34.1 Sustainable Physical Labour

Physical work is limited by capacity that is expended by effort and restored by rest, and anyone who has done sustained manual labour knows that the useful measure is the output over a day or a season and not the output of the first hour. An effort pattern that front-loads expenditure produces a high initial rate and a collapse, and a steady pattern produces less at the start and more in total. The point is familiar, and it is stated here because the model of the chapter is a formalization of it that applies equally to intellectual work, where the same distinction is often not made.

## 34.2 A Simple Dynamical Model

Let $E(t)$ be available capacity, $r$ a constant rate of recovery and $c(u)$ the rate at which effort $u \geq 0$ expends capacity, so that $\dot{E} = r - c(u)$, subject to the constraint that capacity never falls below a floor $E_{\min}$. The productive output over a horizon $T$ is $W_T = \int_0^T u(t)\,dt$. If expenditure is a convex power of effort, $c(u) = u^p$ with $p > 1$, then doing the same work at twice the rate costs more than twice as much, and the model gives the exact form of the penalty. The convexity assumption is the substantive modelling choice. It encodes that effort is subject to increasing marginal cost, which is a standard feature of physical exertion and is plausible for sustained concentration, although the monograph does not claim it as an empirical law of the latter. The model is not a physiological theory. It is a simple model in which the question of the best pacing has a definite answer.

## 34.3 Local Acceleration and Global Degradation

Within the model, increasing effort at a moment raises the output rate at that moment and lowers the capacity available later, and with a fixed total budget the net effect on

total output is negative. This is the content of the proposition on constant pacing: the maximum of total output for a fixed capacity budget is attained by constant effort, and any unequal distribution of the same expenditure yields strictly less. The phenomenon of local acceleration and global degradation is a consequence of convexity and not of any particular scale of time. The corollary on bursts makes it explicit by comparing the output of an effort concentrated in a fraction of the available time to that of constant effort.

## 34.4 Intellectual Pacing

Intellectual work differs from physical work in the observability of its output, since a period of concentrated reasoning may produce nothing visible and a later period may produce a great deal. The model of the chapter treats output as the integral of effort and so does not capture this. It applies to the part of intellectual work that is limited by sustained concentration, and for that part it gives the same advice as for physical work: a rate of effort that can be sustained produces more over a long horizon than a rate that cannot. The qualification should be kept in view, because a model that is accurate for sustained concentration may be misleading for the part of discovery that depends on rare insight.

## 34.5 Polemical Expenditure

Expenditure of capacity on argument consumes the same budget as expenditure on production. If a fraction $\beta$ of the capacity budget is spent on controversy, the maximal output over the horizon falls by the factor $(1-\beta)^{1/p}$, as the allocation proposition below shows. The factor is mild when $\beta$ is small and $p$ is large, and it is substantial when a large fraction of the budget is committed. This is a statement about the cost side only, and it does not say that controversy has no benefit. It says that the benefit must be weighed against a cost that has a definite form.

## 34.6 Sustainable Inquiry as a Form of Independence

A pattern of inquiry that does not exhaust the person who pursues it can be continued indefinitely, whereas one that depends on repeated exhaustion depends on the availability of recovery, which may be interrupted by circumstances outside the person’s control. In this sense sustainable pacing is a condition of independence, since it reduces the dependence of the work on external support for recovery. The connection to the temporal sovereignty of the next chapter is that a person who controls the schedule can pace the work, and a person who does not cannot.

## 34.7 Formal Statements

**Definition 34.1** (Capacity Dynamics and Output)**.** Capacity $E$ satisfies $\dot{E}(t) = r - c(u(t))$ with a constant recovery rate $r \geq 0$, effort $u(t) \geq 0$ and the constraint $E(t) \geq E_{\min}$ for $0 \leq t \leq T$. The productive output is $W_T = \int_0^T u(t)\,dt$. Write $B = E(0) - E_{\min} + rT$ for the capacity budget.

**Proposition 34.2** (Constant Pacing Under Convex Expenditure)**.** *Let $c(u) = u^p$ with $p > 1$. Every admissible effort $u$ satisfies $\int_0^T u^p\,dt \leq B$ and hence $W_T \leq T^{1-1/p}B^{1/p}$, and the constant effort $u^* = (B/T)^{1/p}$ is admissible and attains the bound. Equality holds for an admissible $u$ if and only if $u$ is almost everywhere equal to $u^*$.*

*Proof.* Integrating the dynamics, $E(T) = E(0) + rT - \int_0^T u^p\,dt \geq E_{\min}$ gives $\int_0^T u^p\,dt \leq B$. By Hölder's inequality with exponents $p$ and $p/(p-1)$, $\int_0^T u\,dt \leq T^{1-1/p}\big(\int_0^T u^p\,dt\big)^{1/p} \leq T^{1-1/p}B^{1/p}$, with equality in the first step exactly when $u^p$ is constant almost everywhere and in the second when $\int u^p = B$, that is, for $u = u^*$ almost everywhere. For admissibility of $u^*$, the expenditure up to time $t$ is $Bt/T = (E(0) - E_{\min})t/T + rt \leq E(0) - E_{\min} + rt$ for $t \leq T$, so $E(t) \geq E_{\min}$ throughout. The output of $u^*$ is $T(B/T)^{1/p} = T^{1-1/p}B^{1/p}$. □

**Corollary 34.3** (Burst Penalty)**.** *If effort vanishes outside a set of times of total length $\alpha T$ with $0 < \alpha \leq 1$, every admissible effort satisfies $W_T \leq \alpha^{1-1/p}T^{1-1/p}B^{1/p}$, which is $\alpha^{1-1/p}$ times the constant-pacing output, and strictly less than it for $\alpha < 1$.*

*Proof.* Apply Hölder's inequality on the set of length $\alpha T$ where $u$ may be nonzero, giving $\int u \leq (\alpha T)^{1-1/p}(\int u^p)^{1/p} \leq (\alpha T)^{1-1/p}B^{1/p}$. Since $1 - 1/p > 0$, the factor $\alpha^{1-1/p}$ is less than one for $\alpha < 1$. □

**Proposition 34.4** (Allocation of the Budget)**.** *If a fraction $\beta \in [0, 1)$ of the capacity budget is committed to other expenditure, the maximal productive output is $(1-\beta)^{1/p}T^{1-1/p}B^{1/p}$, that is, $(1-\beta)^{1/p}$ times the output without that commitment.*

*Proof.* The productive effort has the budget $(1-\beta)B$, and Proposition 34.2 with this budget gives the bound, attained by constant effort. □

*Remark* 34.5. The model is explicit about the distinction between racing to maximize immediate output and pacing to maximize sustained output. It is not a complete physiological or psychological theory, and the assumptions of constant recovery, power-law expenditure and output equal to the integral of effort are modelling choices.

# Chapter 35

# Computational Dispossession and Temporal Sovereignty

## 35.1 Material Dispossession

Material dispossession is the loss of the means by which work is done: access to tools, data, software and the computational infrastructure on which an artifact depends. In the accounting of Chapter 26 it appears as the access cost that rises until reuse is no longer advantageous, or as the inaccessibility of a certificate. The formal treatment there requires no assumption about how the loss occurs and shows what follows from it. This chapter turns to a second form of loss, that of time, which has the same structure: a quantity that the work requires and that is reduced by actions of other parties, with a quantitative effect on the attainable output.

## 35.2 Temporal Dispossession

Temporal dispossession is the reduction of the time over which a person can work without compulsory interruption. The available working horizon is the total time less the intervals taken by interruptions, and each interruption takes at least a minimum interval that includes the time to restore the activity. The model is deliberately simple. It treats interruptions as removing time and as inflicting a continuation loss, a reduction in the reusable structure of the working state, and it asks what follows for the output attainable under the pacing model of Chapter 34. The answer is that the loss of time costs output sublinearly in the model of the preceding chapter, because effort can be concentrated in the remaining time at a penalty given by the burst corollary, but that the penalty is real and that it is imposed by the schedule and not by the person.

## 35.3 Attention as a Condition of Productive Capacity

Attention is not a resource that can be spent in arbitrary pieces. A task that requires sustained concentration has a minimum duration below which no progress is made, and an interruption that breaks the duration wastes the part already spent. In the model this is accounted for by the continuation loss, which measures the working structure lost at the interruption and is reconstructed at a cost. The conclusion that an interruption costs more than its duration is thus built into the model, and it is a stronger consequence than the arithmetic of lost time.

## 35.4 The Cost of Compulsory Reorientation

An interruption that is compulsory, in the sense that the person cannot decline it without a cost of their own, imposes the restoration cost without consent. The model of temporal sovereignty represents this as a restriction of the set of admissible schedules. The person who controls the schedule chooses among schedules that include uninterrupted periods, and the person subject to compulsory interruptions chooses among a subset. The proposition on feasible-set contraction states the elementary consequence: the best attainable value under any objective cannot increase when the feasible set is restricted.

## 35.5 Intellectual Independence as Control over Continuation

Independence in intellectual work is, in the model, control over continuation: the ability to resume the work from the state at which it stood. It depends on access to the tools, the time and the working state, and it is lost by their removal. The formal results of this part do not establish that any particular person or institution is dispossessed. They show what the loss of time and access costs in a stated model, and they connect the loss to the accountability and control relations of Part XI, since a party that cannot continue a work cannot maintain it.

## 35.6 Formal Statements

**Definition 35.1** (Effective Working Time)**.** Let $T$ be the total available time and let interruptions occupy pairwise disjoint intervals of lengths $\tau_1, \dots, \tau_N$ within $[0, T]$, each including the time of restoration. The effective working time is $T_{\text{eff}} = T - \sum_{j=1}^{N} \tau_j$.

**Proposition 35.2** (Interruption Accumulation)**.** *If $\tau_j \geq \tau_{\min} > 0$ for all $j$, then $T_{\text{eff}} \leq T - N\tau_{\min}$. Consequently $N \leq T/\tau_{\min}$ when $T_{\text{eff}} \geq 0$, and if interruptions occur at rate $\lambda = N/T$ then $T_{\text{eff}}/T \leq 1 - \lambda\tau_{\min}$.*

*Proof.* Summing $\tau_j \geq \tau_{\min}$ gives $\sum_j \tau_j \geq N\tau_{\min}$, hence $T_{\text{eff}} \leq T - N\tau_{\min}$. The disjoint intervals lie in $[0, T]$, so $T_{\text{eff}} \geq 0$ and $N\tau_{\min} \leq T$. Dividing by $T$ gives the last claim. □

**Definition 35.3** (Continuation Loss)**.** Let $K(x) \geq 0$ measure the reusable working structure present in the state $x$. An interruption at time $t_j$ produces the continuation loss $\Delta K_j = K(x_{t_j^-}) - K(x_{t_j^+})$, where the post-interruption state is measured before reconstruction.

**Hypothesis 35.4** (Restoration Cost Proportional to Loss)**.** *The interval $\tau_j$ of the $j$th interruption satisfies $\tau_j \geq \rho\,\Delta K_j$ for a constant $\rho > 0$, so that restoration time grows with continuation loss.*

**Corollary 35.5** (Loss-Dependent Time Bound)**.** *Under Hypothesis 35.4, $T_{\text{eff}} \leq T - \rho \sum_j \Delta K_j$.*

*Proof.* Sum the inequality $\tau_j \geq \rho \Delta K_j$ and subtract from $T$. □

**Corollary 35.6** (Output Under Interruption)**.** *In the model of Proposition 34.2, if effort vanishes during the interruptions, the output satisfies $W_T \leq (T_{\text{eff}}/T)^{1-1/p}\, T^{1-1/p} B^{1/p}$, with equality attainable by constant effort on the uninterrupted time at the rate $(B/T_{\text{eff}})^{1/p}$ when this is admissible.*

*Proof.* Apply Corollary 34.3 with $\alpha = T_{\text{eff}}/T$. For the equality, constant effort $u = (B/T_{\text{eff}})^{1/p}$ on a set of measure $T_{\text{eff}}$ has output $T_{\text{eff}}^{1-1/p} B^{1/p}$, which is the bound, and it is admissible whenever the capacity constraint holds along the schedule. □

**Definition 35.7** (Temporal Sovereignty)**.** Let $\mathcal{U}_{\text{possible}}$ be the set of schedules that are possible and $\mathcal{U}_{\text{self}} \subseteq \mathcal{U}_{\text{possible}}$ the set of admissible user-controlled schedules. Compulsory external interruptions restrict $\mathcal{U}_{\text{self}}$.

**Proposition 35.8** (Feasible-Set Contraction)**.** *If $\mathcal{U}_2 \subseteq \mathcal{U}_1$, then for every objective $J$, $\sup_{u \in \mathcal{U}_2} J(u) \leq \sup_{u \in \mathcal{U}_1} J(u)$. If the supremum over $\mathcal{U}_1$ is attained at some $u_0 \notin \mathcal{U}_2$ and is not attained in $\mathcal{U}_2$ at the same value, the inequality can be strict.*

*Proof.* Every element of $\mathcal{U}_2$ is in $\mathcal{U}_1$, so the supremum over $\mathcal{U}_1$ is at least the value of every element of $\mathcal{U}_2$. For strictness take $\mathcal{U}_1 = \{u_0, u_1\}$, $\mathcal{U}_2 = \{u_1\}$ and $J(u_0) > J(u_1)$. □

**Proof Obligation 35.9.** *The empirical claims suggested by this part, that particular systems manufacture urgency, that interruptions have the cost structure assumed, and that people are deprived of the time or means of work by particular practices, are not established by the propositions above. They are conditional on the models and require evidence.*

# Part XIII

# Generalizing the Mathematics of Obstruction

# Chapter 36

# Local Transitions and Global Admissibility

## 36.1 Local Operations

A distributed system is built from components, each of which has a set of states and performs local operations that depend on its own state and on what it receives from its neighbours. The configuration space of the whole is the product of the component state spaces, and a configuration is admissible if it satisfies the constraints that the system as a whole must respect: consistency of replicated data, conservation laws, protocol invariants. The same structure occurs in the preceding parts in a different guise. A clause sum is a sum over assignments of bits of a product of local weights, and its terms are the admissible global configurations weighted by local factors. The present chapter abstracts the structure and asks what can be inferred about global admissibility from what is visible to each component.

## 36.2 Global Compatibility

Global admissibility is a property of the configuration as a whole, and it is not in general recoverable from local admissibility, the set of states of each component that occur in some admissible configuration. The simplest example is a pair of components that must agree: each state of each component is locally admissible, and half of the combinations are not. The example is elementary, and it is the structure behind every failure of local checks to establish global properties. A system in which local admissibility suffices has a product structure, and the proposition below shows that this is the only case in which it suffices.

## 36.3 The Complexity of Compatibility

The cost of deciding whether a configuration is admissible, the cost of finding one, and the cost of representing the set of admissible configurations are three different quantities.

The first can be low where the second is high, as in the relation between verification and search of Chapter 1. The third is the one measured by tensor rank in this monograph: the indicator of the admissible set, viewed as a tensor, has flattening ranks that measure how much information must pass between two parts of the system for their joint state to be admissible. A product set has rank one across every cut, and a set that couples the two sides has higher rank. The rank across a cut is therefore the representational cost of the coupling.

## 36.4 Tensor Representations of Admissibility

For finite state spaces the indicator function of the admissible set is a tensor whose entry at a configuration is one if the configuration is admissible and zero otherwise. Its flattenings across bipartitions of the components are 0-1 matrices, and their ranks, taken over a field, are the invariants used throughout the geometric argument. The proposition below identifies exactly when the rank across a cut is one: when the admissible set factors across the cut. In this sense the flattening rank is a graded measure of failure of the admissible set to factor, and the integer $\log_2$ of the rank is the number of bits of interface that a linear representation of the coupling requires, as developed in the next chapter.

## 36.5 Limits of the Representation

The tensor representation records the admissible set and nothing about the cost of the local operations, the dynamics by which a system moves between admissible configurations, or the physical medium of the interfaces. It is a static description. The rank over a field can depend on the characteristic, since a 0-1 matrix can have different ranks over different fields, and the monograph uses it where the argument requires only that it be a rank of some fixed field. For the structural statements below the characteristic is immaterial.

## 36.6 Formal Statements

**Definition 36.1** (Configuration Space and Admissibility)**.** Let $V$ be a finite set of components with finite state sets $X_v$. The configuration space is $X = \prod_{v \in V} X_v$, the globally admissible set is a subset $\mathcal{A} \subseteq X$, and for $S \subseteq V$ the locally admissible set is $\mathcal{A}_S = \pi_S(\mathcal{A})$, where $\pi_S$ is the coordinate projection.

**Proposition 36.2** (Local Admissibility Is Insufficient)**.** *For every $\mathcal{A}$, $\mathcal{A} \subseteq \prod_{v \in V} \mathcal{A}_{\{v\}}$, and the inclusion can be strict. Equality holds if and only if $\mathcal{A}$ is a product set $\prod_v A_v$.*

*Proof.* A configuration in $\mathcal{A}$ has each coordinate in the corresponding projection, so it lies in the product. Strictness: let $V = \{1, 2\}$, $X_1 = X_2 = \{0, 1\}$ and $\mathcal{A} = \{(0, 0), (1, 1)\}$. Both projections are $\{0, 1\}$, and the product $\{0, 1\}^2$ contains the inadmissible configurations $(0, 1)$ and $(1, 0)$. If $\mathcal{A} = \prod_v A_v$ with nonempty $A_v$, then $\mathcal{A}_{\{v\}} = A_v$ and equality holds. Conversely if equality holds, $\mathcal{A} = \prod_v \mathcal{A}_{\{v\}}$ is a product set. □

**Definition 36.3** (Admissibility Tensor)**.** For finite state sets the admissibility tensor is $T_{\mathcal{A}} = \mathbf{1}_{\mathcal{A}} \in \bigotimes_v \mathbb{F}^{X_v}$, with entry 1 at admissible configurations and 0 elsewhere. For $\varnothing \neq S \subsetneq V$ write $r_S(T_{\mathcal{A}})$ for the rank of its flattening.

**Proposition 36.4** (Product Admissibility)**.** *If $\mathcal{A} = \prod_v A_v$ with every $A_v$ nonempty, then $T_{\mathcal{A}} = \bigotimes_v \mathbf{1}_{A_v}$ and $r_S(T_{\mathcal{A}}) = 1$ for every $\varnothing \neq S \subsetneq V$.*

*Proof.* The indicator of a product set is the product of the indicators, so $\mathbf{1}_{\mathcal{A}}(x) = \prod_v \mathbf{1}_{A_v}(x_v)$. The flattening is then the outer product of the vector $\bigotimes_{v \in S} \mathbf{1}_{A_v}$ with the vector $\bigotimes_{v \notin S} \mathbf{1}_{A_v}$, both nonzero, so its rank is one. □

**Proposition 36.5** (Rank One Characterizes Factorization)**.** *Let $\mathcal{A} \neq \varnothing$ and $\varnothing \neq S \subsetneq V$. Then $r_S(T_{\mathcal{A}}) = 1$ if and only if $\mathcal{A} = \pi_S(\mathcal{A}) \times \pi_{S^c}(\mathcal{A})$. Moreover $r_S(T_{\mathcal{A}}) = 1$ for every singleton $S = \{v\}$ if and only if $\mathcal{A}$ is a product set.*

*Proof.* The flattening is the 0-1 matrix $M$ with $M_{a,b} = 1$ iff $(a, b) \in \mathcal{A}$, for $a$ a configuration of $S$ and $b$ of $S^c$. Its rank is one iff all its nonzero rows are proportional. Two nonzero 0-1 rows that are proportional have the same support and the proportionality constant 1, so they are equal. Hence rank one holds iff all nonzero rows are equal, that is, iff there is a set $B \subseteq X_{S^c}$ with $\mathcal{A} = \{a : \text{row } a \neq 0\} \times B = \pi_S(\mathcal{A}) \times \pi_{S^c}(\mathcal{A})$. For the second claim a product set has this property for every $S$ by Proposition 36.4. Conversely suppose $\mathcal{A} = \mathcal{A}_{\{v\}} \times \pi_{V \setminus v}(\mathcal{A})$ for each $v$, and let $x$ have $x_v \in \mathcal{A}_{\{v\}}$ for all $v$. Start from any $y \in \mathcal{A}$. Enumerating $V = \{v_1, \ldots, v_n\}$, replace $y_{v_1}$ by $x_{v_1}$: the result is in $\mathcal{A}$, since $\mathcal{A} = \mathcal{A}_{\{v_1\}} \times \pi_{V \setminus v_1}(\mathcal{A})$ allows any value in $\mathcal{A}_{\{v_1\}}$ at $v_1$. Repeating for $v_2, \ldots, v_n$ yields $x \in \mathcal{A}$. So $\prod_v \mathcal{A}_{\{v\}} \subseteq \mathcal{A}$, and with Proposition 36.2 the set is a product. □

*Example* 36.6. For $\mathcal{A} = \{(0, 0), (1, 1)\}$ the flattening is the $2 \times 2$ identity matrix, of rank two over every field, which is the coupling that local admissibility fails to see.

# Chapter 37

# Gluing and the Obstruction to Independent Reconstruction

## 37.1 Local Sections

A local section is a piece of a description that is valid over part of a system: a state of some components, a fragment of a proof, a local view of a computation. A cover is a family of the parts over which local sections are given. The vocabulary is that of sheaf theory, used here in the elementary form needed to state which local data can be assembled into a global object. The question that motivates it is the one of Part X in another form: a party who holds only local pieces may be unable to reconstruct the whole, and the obstruction can be a matter of cost or of impossibility.

## 37.2 Agreement on Overlaps

Two local sections are compatible if they agree where their domains overlap. Compatibility is a necessary condition for the existence of a global section restricting to both, and for sheaves it is sufficient. For presheaves and for constrained realization systems it is not, and the failure is the content of the previous chapter: the sections of the admissible set over single components are compatible, because the components do not overlap, and they need not assemble to an admissible configuration. The proposition below states this in sheaf language, so that the abstract notion of gluing and the elementary example are seen to be the same.

## 37.3 Global Reconstruction

Reconstruction is the passage from local sections to a global section. In a sheaf it exists and is unique whenever the sections are compatible. In general it may fail to exist, may exist in several ways, and, where it exists, has a cost that depends on the procedure. The gluing

expenditure of the formal statements is the least cost among admissible procedures, and it is taken to be infinite when none exists, in the convention of Chapter 1 for the infimum over an empty set.

## 37.4 Tensor Rank and Gluing Complexity

When the global object is a tensor and the local sections are the parts held on each side of a cut, the interface through which the two sides communicate is a space of some dimension, and a linear interface of dimension $r$ realizes the tensor if it is a sum of $r$ terms, each a product of a tensor on one side and a tensor on the other. The flattening rank is the least such $r$, so that the logarithm of the rank is a lower bound on the number of bits that a linear interface must carry. The bound is attained by a rank factorization, and it is the exact measure of the cost for a single cut. For a network with many cuts, the argument of Parts V to VII is concerned with what the bounds across many cuts jointly imply.

## 37.5 Obstruction Versus Nonexistence

The distinction between obstruction and nonexistence is the one the monograph has used from the start. In a finite setting a rank factorization always exists, so that gluing across a single cut is never impossible and always has a definite cost, namely the rank. An obstruction here is a lower bound on a cost and not a proof that no reconstruction exists. Nonexistence, by contrast, occurs for the presheaf of admissible configurations when the compatible local sections do not assemble, and the failure is independent of any cost. Confusing the two leads to claims of impossibility where only expense has been shown, or to claims of expense where nothing can be done.

## 37.6 Formal Statements

**Definition 37.1** (Local Sections and Compatibility)**.** Let $\mathcal{F}$ be a presheaf on a finite set $V$, assigning to each $U \subseteq V$ a set $\mathcal{F}(U)$ and to $U' \subseteq U$ a restriction $\mathcal{F}(U) \to \mathcal{F}(U')$, functorially. For a cover $\{U_i\}$ of $V$, local sections are $s_i \in \mathcal{F}(U_i)$, and they are compatible if $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$ for all $i, j$. A global realization is $s \in \mathcal{F}(V)$ with $s|_{U_i} = s_i$ for all $i$.

**Definition 37.2** (Gluing Expenditure)**.** Let $\Gamma(\mathbf{s})$ be the set of admissible global reconstruction procedures for the local sections $\mathbf{s} = (s_i)$ and $C$ the cost of a procedure. The gluing expenditure is $C_G(\mathbf{s}) = \inf_{\gamma \in \Gamma(\mathbf{s})} C(\gamma)$, with $C_G(\mathbf{s}) = \infty$ if $\Gamma(\mathbf{s}) = \varnothing$.

**Proposition 37.3** (Admissibility Presheaf)**.** *For $\mathcal{A} \subseteq X = \prod_v X_v$, the assignment $\mathcal{F}_{\mathcal{A}}(U) = \pi_U(\mathcal{A})$ with restriction by coordinate projection is a presheaf. It is separated: for a cover of $V$ a global realization is unique if it exists. Gluing of compatible sections over the cover by singletons holds if and only if $\mathcal{A}$ is a product set.*

*Proof.* If $U' \subseteq U$ then $\pi_{U'} = \pi_{U'} \circ \pi_U$ maps $\pi_U(\mathcal{A})$ into $\pi_{U'}(\mathcal{A})$, so restriction is defined and functorial. A global section is a tuple in $X$ and is determined by its coordinates, which are restrictions to the members of a cover of $V$, so a global realization is unique. The sections over the singletons are compatible, since distinct singletons have empty overlap, and they glue exactly when every tuple $(s_v)$ with $s_v \in \mathcal{A}_{\{v\}}$ lies in $\mathcal{A}$, that is, when $\prod_v \mathcal{A}_{\{v\}} \subseteq \mathcal{A}$. By Proposition 36.2 this holds iff $\mathcal{A}$ is a product set. □

**Definition 37.4** (Boundary Complexity)**.** For a nonzero tensor $T$ and a bipartition $S \mid S^c$ of its parties, $B_S(T) = \log_2 r_S(T)$.

**Proposition 37.5** (Linear Interface Lower Bound)**.** *If $T = \sum_{i=1}^{r} a_i \otimes b_i$ with each $a_i$ a tensor on $S$ and each $b_i$ a tensor on $S^c$, then $r \geq r_S(T)$, so that $\log_2 r \geq B_S(T)$. Conversely $T$ has such an expression with $r = r_S(T)$, so $r_S(T)$ is the least dimension of a linear interface across the cut.*

*Proof.* The flattening of $T$ is $\sum_{i=1}^{r} \operatorname{vec}(a_i) \operatorname{vec}(b_i)^\top$, a sum of $r$ matrices of rank at most one, so its rank is at most $r$. Conversely a matrix of rank $\rho$ is a sum of $\rho$ rank-one matrices, by rank factorization, and each rank-one matrix is an outer product of a vector on the rows and a vector on the columns. □

**Corollary 37.6** (Obstruction Is Not Nonexistence)**.** *For a finite tensor the gluing across a single cut always exists, with interface dimension $r_S(T)$, so that the lower bound $\log_2 r_S(T)$ is a statement about cost. By contrast, for $\mathcal{A} = \{(0,0),(1,1)\}$ the sections $s_1 = 0$ and $s_2 = 1$ over the singletons are compatible and have no global realization, in $\mathcal{F}_{\mathcal{A}}$.*

**Hypothesis 37.7** (Rank Bounds Gluing Expenditure)**.** *For suitable reconstruction models, the boundary complexity $B_S(T)$ lower-bounds a specified gluing expenditure $C_G$. Such a relation requires an explicit reconstruction model, in which the cost of a procedure is related to the dimension of the interface it uses, and cannot be assumed universally.*

# Chapter 38

# Admissibility-Preserving Abstraction

## 38.1 Quotient Maps and Lost Distinctions

An abstraction is a map that forgets some distinctions: a quotient of configurations, a projection of states, an evaluation of polynomials on a subset of inputs. The map identifies objects that it does not separate, and anything defined on the original objects is either constant on the identified classes, in which case it survives the abstraction, or not. The Boolean evaluation map is the central example of the monograph, and the general statement is a criterion that has a short proof and wide application: a quantity survives an abstraction exactly when it is constant on the fibers.

## 38.2 Preservation of Outputs

An abstraction preserves an output if the output can be computed from the abstracted object. For a function $C$ on the objects, this is the existence of a function on the abstracted objects that composes with the abstraction to give $C$, and it is equivalent to constancy of $C$ on fibers. The equivalence is useful because the second form is checkable on the original objects, by exhibiting two objects in the same fiber with different values of $C$, and the first form, which is what one wishes to know, is existential.

## 38.3 Preservation of Obstruction

An obstruction is a lower bound on a cost. Whether the obstruction survives the abstraction depends on the abstract cost, the lower envelope over the fiber, and the lower envelope is at most the cost of any representative. A lower bound proved for one representative bounds the lower envelope only if it holds for every representative in the fiber, which is what the proof of the lower bound would have to establish. The upper envelope and the distortion measure

how far the representatives differ in cost, and a large distortion means that a bound on one representative says little about the abstraction. A cost is exactly preserved if it is constant on fibers, and the case of Chapter 3 is a quotient in which the cost is far from constant.

## 38.4 The Boolean Evaluation Example

The Boolean evaluation map $E$, sending a polynomial to its function on the cube, has fibers that are the cosets of the ideal generated by $x_i^2 - x_i$. Two quantities defined on polynomials can be examined against it. The extraction of the part of degree one in every block of variables is not constant on the fibers, and in the strong sense of Chapter 22: the values of the extraction on a single fiber exhaust the whole space of possible values. The circuit size is also not constant on the fibers, and its upper envelope is infinite. The two examples show the extremes of distortion, and they make precise the sense in which the argument of the source about circuits applies to formal polynomials and not to their Boolean evaluations. A concrete second example, from software dependencies, shows an abstraction that forgets direction and within-category edges.

## 38.5 Sufficient Abstractions for Complexity Analysis

An abstraction is sufficient for a cost if the cost descends through it. The sufficient abstractions for a given cost are those whose fibers lie within the level sets of the cost, and the coarser the abstraction, the fewer the costs that descend through it. Choosing an abstraction for analysis is therefore choosing which costs can be computed from the abstract object, and a bound proved on an abstraction that is not sufficient for the cost of interest is a bound on something else. The proposition on composition below records the monotonicity of this relation.

## 38.6 Formal Statements

**Definition 38.1** (Abstraction and Descent)**.** An abstraction is a surjection $q : X \to Y$. A function $C : X \to \mathbb{R} \cup \{\infty\}$ descends through $q$ if there is $\bar{C} : Y \to \mathbb{R} \cup \{\infty\}$ with $C = \bar{C} \circ q$.

**Theorem 38.2** (Fiber Criterion)**.** *$C$ descends through $q$ if and only if $q(x) = q(x')$ implies $C(x) = C(x')$.*

*Proof.* If $C = \bar{C} \circ q$ and $q(x) = q(x')$, then $C(x) = \bar{C}(q(x)) = \bar{C}(q(x')) = C(x')$. Conversely suppose $C$ is constant on fibers. Since $q$ is surjective, each $y \in Y$ has a preimage $x$, and set $\bar{C}(y) = C(x)$. The value does not depend on the choice of $x$ by constancy on the fiber, so $\bar{C}$ is well defined, and $\bar{C}(q(x)) = C(x)$ by construction. □

**Definition 38.3** (Envelopes and Distortion)**.** $C_-(y) = \inf_{x \in q^{-1}(y)} C(x)$ and $C_+(y) = \sup_{x \in q^{-1}(y)} C(x)$. When both are finite, the obstruction distortion is $\Delta_C(y) = C_+(y) - C_-(y)$.

**Proposition 38.4** (Envelope Bounds)**.** *$C_-(q(x)) \le C(x) \le C_+(q(x))$ for every $x$, and $C$ descends through $q$ if and only if $C_- = C_+$ on $Y$, in which case $C = C_- \circ q$.*

*Proof.* The first claim holds because $x$ lies in the fiber over $q(x)$. If $C_-(y) = C_+(y)$, every value of $C$ on the fiber lies between them and so equals the common value, and $C$ is constant on fibers. Conversely constancy gives equal infimum and supremum. □

**Proposition 38.5** (Monotonicity of Descent)**.** *Let $q : X \to Y$ and $r : Y \to Z$ be abstractions. If $C$ descends through $r \circ q$, then it descends through $q$. If $C$ descends through $q$ with $\bar{C}$, it descends through $r \circ q$ if and only if $\bar{C}$ descends through $r$.*

*Proof.* A fiber of $q$ is contained in a fiber of $r \circ q$, so constancy on the larger fibers gives constancy on the smaller. For the second claim, $C = \bar{C} \circ q$ and $C$ is constant on the fibers of $r \circ q$ iff $\bar{C}$ is constant on the fibers of $r$, since $q$ is surjective, and Theorem 38.2 applies. □

**Proposition 38.6** (Boolean Evaluation Has Maximal Distortion)**.** *Fix $n \ge 1$ and a partition of the variables into nonempty blocks, let $X = \mathbb{F}[x_1, \dots, x_n]$, $Y = \mathbb{F}^{\{0,1\}^n}$ and $q = E$. For every fiber $f + I_B$ the set $M(f + I_B)$ equals the space of all polynomials of degree exactly one in each block. Consequently, for $\chi(g) = 1$ if $M(g) \neq 0$ and $\chi(g) = 0$ otherwise, $\chi_- = 0$ and $\chi_+ = 1$ on every fiber. For the circuit size $L$, $L_+(y) = \infty$ on every fiber and $L_-(y) = L_B(y)$.*

*Proof.* By Proposition 22.6, $M(I_B)$ is the space of all polynomials of degree one in each block, and $M(f + e) = M(f) + M(e)$ ranges over $M(f)$ plus this space, which is the same space. It contains 0 and a nonzero polynomial, so $\chi$ takes both values on every fiber. For the size, let $f$ have degree $\delta$ and take $g_k = f + (x_1^2 - x_1)x_1^k$ for $k \ge \delta$. Then $g_k \in f + I_B$ and $\deg g_k = k + 2$. A circuit of $L$ entries computes a polynomial of degree at most $2^L$, so $L(g_k) \ge \log_2(k + 2) \to \infty$, and $L_+ = \infty$. The lower envelope is $L_B$ by definition. □

*Remark* 38.7. Proposition 38.6 is the general framework's form of Chapter 22: the extraction value is completely undetermined by the Boolean equivalence class, and the circuit size is determined by it only through the lower envelope, which no lower bound on a single representative controls.

**Definition 38.8** (Dependency Shadow)**.** Let $G = (V, E)$ be a directed graph, possibly with a distinguished element, and let $\lambda : V \to \Lambda$ be a classification of the vertices into categories. The shadow edge set is $E_{\text{shadow}} = \{(u, v) \in E : \lambda(u) \neq \lambda(v)\}$, the undirected shadow edge set is $F_{\text{shadow}} = \{\{u, v\} : (u, v) \in E_{\text{shadow}} \text{ or } (v, u) \in E_{\text{shadow}}\}$, and the shadow graph is $H = (V, F_{\text{shadow}})$. The map $\sigma : G \mapsto H$ is an abstraction onto the set of shadows.

**Proposition 38.9** (What the Shadow Preserves and Loses)**.** *Two directed graphs on $V$ have the same shadow if and only if they have the same set of unordered pairs joined by a cross-category edge in either direction. The following descend through $\sigma$: the number of shadow edges, the cut capacity $B_H(S) = \sum_{e \in \delta_H(S)} b_e$ for any assignment of capacities to unordered pairs, and the connected components of $H$. The reachability relation of $G$ does not: there are $G, G'$ with the same shadow and a pair $(a, c)$ such that $c$ is reachable from $a$ in $G$ and not in $G'$.*

*Proof.* The first statement is the definition of $F_{\mathrm{shadow}}$. The first three quantities are functions of $F_{\mathrm{shadow}}$ alone, so they are constant on fibers. For reachability let $V = \{a, b, c\}$ with $\lambda(a) \neq \lambda(b) = \lambda(c)$, $G$ with edges $a \to b$ and $b \to c$, and $G'$ with only $a \to b$. The edge $b \to c$ is within a category and does not appear in the shadow, so both graphs have $F_{\mathrm{shadow}} = \{\{a, b\}\}$, while $c$ is reachable from $a$ in $G$ and not in $G'$. Reversing the edge $a \to b$ also leaves the shadow unchanged and changes reachability. □

*Remark* 38.10. In the notation of Definition 38.3 the indicator of reachability of $c$ from $a$ has lower envelope 0 and upper envelope 1 on the fiber of $G$, so the distortion is maximal for that quantity, while the cut capacities have distortion zero. A statistic computed on the shadow can therefore be informative about the structure of cross-category coupling and uninformative about propagation along directed paths. This is the sense in which a graph abstraction can preserve certain structural statistics and destroy information needed for security properties.

# Chapter 39

# Distributed Computation and Boundary Resources

## 39.1 Modular Computational Architectures

A modular architecture divides a computation among components that exchange information across interfaces. The partition into modules is chosen for reasons of design, and its consequences for what the architecture can compute are determined by the interfaces: a module sees the rest of the system only through the information that crosses its boundary. The formal model represents an architecture as a graph with capacities on the edges, and a computation as a tensor to be realized by a network on the graph. The question asked is which tensors are realizable, and the answer is given by the rank across cuts.

## 39.2 Boundary Information

The boundary of a set of modules is the set of edges that connect it to the rest. The information that can cross the boundary is limited by the capacities of those edges, and the information that must cross it, for a given computation, is measured by the flattening rank of the tensor across the corresponding bipartition. The first is a property of the architecture and the second of the computation. A computation is feasible on an architecture only if the second does not exceed the first across any cut, and the proposition below states this as an inequality for tensor networks, with the interface feasibility corollary in the contrapositive.

## 39.3 Network Cuts and Communication Constraints

The cut capacity of a set of modules is the sum, over the boundary edges, of the capacities. With the convention that an edge carrying a wire of dimension $q$ has capacity $\log_2 q$ bits, the capacity of a cut is the logarithm of the product of the dimensions, and the rank bound says that the logarithm of the flattening rank of the tensor realized by a network is at most

the capacity of any cut. This is the communication-complexity form of the statement, in which the flattening rank measures what must be communicated and the capacity what can be. The boundary deficit, the amount by which the requirement exceeds the capacity, is a certificate of infeasibility within the model.

## 39.4 Complete Dependency Structures

A complete dependency structure is one in which every pair of components is coupled, as in the complete network of Chapter 14, where every pair of nodes shares a bundle of $\ell$ bits. In such a structure every balanced cut has a capacity proportional to $\ell k^2$, and this is the source of the hardness of the network polynomial: its tensor has high rank across every balanced cut, so that any decomposition into few pieces must route a great deal across some cut. The example of a software dependency graph with an interface dimension on every shadow edge is of the same kind, and for it the cut rank is given exactly by the product formula of Chapter 9.

## 39.5 Physical Implementation

The identification of the dimension of a tensor index with a communication capacity is a model. Physical bandwidth and latency interpretations require assumptions that connect tensor dimensions to the protocols, media and costs of an implementation, and these are not supplied by the formal development. The results below are statements about tensor networks, and they apply to a physical system to the extent that the system's interfaces are faithfully represented by wires of the stated dimensions.

## 39.6 The Geometry of Architectural Feasibility

In the geometric argument of Parts V to VII, a spectral point converts the wires of a network into a metric on its nodes, and the cost of realizing a tensor by a network is a sum of lengths. The metric is a property of the pair of the spectral point and the architecture, and feasibility of a network is a statement about whether the lengths allow it. The extension of this geometric viewpoint to architectures in general, where the question would be whether curvature properties of the induced metric constrain feasibility, is a research hypothesis and is stated in the next chapter in the abstract setting.

## 39.7 Formal Statements

**Definition 39.1** (Modular Graph and Cut Capacity)**.** A modular graph is $G = (V, E)$ with capacities $b_e \geq 0$ on the edges. For $S \subseteq V$ the cut capacity is $B_G(S) = \sum_{e \in \partial S} b_e$, where $\partial S$ is the set of edges with exactly one endpoint in $S$.

**Definition 39.2** (Tensor-Network Capacity)**.** A tensor network on $G$ assigns to each vertex a local tensor with open indices and to each edge $e$ a contracted index of dimension $q_e$. The edge capacity is $b_e = \log_2 q_e$, so that $B_G(S) = \log_2 \prod_{e \in \partial S} q_e$.

**Proposition 39.3** (Network Cut Rank)**.** *Let $T$ be the tensor of a tensor network on $G$, with the open indices of each vertex regarded as parties at that vertex, and let $S \subseteq V$. Then $r_{S|S^c}(T) \leq \prod_{e \in \partial S} q_e$, so $\log_2 r_{S|S^c}(T) \leq B_G(S)$.*

*Proof.* Contract first the edges inside $S$ and inside $S^c$. The remaining sum is over the assignments $\alpha$ of the indices of the edges in $\partial S$, and for each $\alpha$ the term is the product of a tensor $T_S^\alpha$ on the open indices at $S$ and a tensor $T_{S^c}^\alpha$ on those at $S^c$. Hence $T = \sum_\alpha T_S^\alpha \otimes T_{S^c}^\alpha$ has $\prod_{e \in \partial S} q_e$ terms, and Proposition 37.5 gives the bound. □

**Corollary 39.4** (Interface Feasibility)**.** *If $B_G(S) < \log_2 r_{S|S^c}(T)$ for some cut, then $T$ is not the tensor of any tensor network on $G$ with these capacities.*

*Proof.* This is the contrapositive of Proposition 39.3. □

**Definition 39.5** (Boundary Deficit)**.** $D_G(T, S) = \max\{0, \log_2 r_{S|S^c}(T) - B_G(S)\}$. A positive deficit certifies infeasibility within the model.

**Proposition 39.6** (Equality for Wire Networks)**.** *Let $T_G = \prod_e W_{q_e}(u_e, v_e)$ be the graph tensor of Chapter 9, with one party at each vertex. Then $\log_2 r_S(T_G) = B_G(S)$ for every $S$, so the boundary deficit of $T_G$ on $G$ is zero for every cut, and the bound of Proposition 39.3 is attained.*

*Proof.* This is Proposition 9.6: $r_S(T_G) = \prod_{e \in \delta(S)} q_e$, whose logarithm is the capacity of the cut. □

*Example* 39.7 (Shadow Graph as a Wire Network)*.* Let $H = (V, F_{\text{shadow}})$ be a shadow graph, with an interface dimension $q_e \geq 2$ assigned to each shadow edge, and let $T_{\text{shadow}} = \prod_{\{u,v\} \in F_{\text{shadow}}} W_{q_{uv}}(u, v)$. By Proposition 39.6, for every bipartition $S \mid S^c$ of the vertices, $\log_2 r_S(T_{\text{shadow}}) = \sum_{e \in \delta_H(S)} \log_2 q_e$. The cut rank therefore measures the total interface capacity of the cross-category couplings that cross the partition, and it is blind to everything that Proposition 38.9 shows the shadow forgets.

**Proof Obligation 39.8.** *The relation of the tensor-network model to physical bandwidth and latency, and the question whether curvature properties of the induced metric constrain architectural feasibility beyond the cut bounds, are not established here.*

# Chapter 40

# A Proposed General Theory of Computational Obstruction

## 40.1 Admissible Transformations

The theory proposed in this chapter abstracts the structure of the argument of Parts IV to VII. In that argument the objects are tensors, the transformations are restrictions, which apply a linear map at each party, and the composition of objects is the tensor product. An admissible transformation is one of the allowed moves in a system of this kind, and what the system can do is described by the preorder that the moves generate. The abstract setting replaces tensors by the elements of a set with a preorder and a composition, and the question is how much of the machinery transfers.

## 40.2 Compositional Structure

The composition operation and its unit make the set a monoid, and the requirement that the preorder be compatible with it, that is, that the composition of related elements be related, makes the whole an ordered monoid. This is what allows the obstruction functionals to be multiplicative. A functional that is monotone for the preorder and multiplicative for the composition gives, from the value on a few elements, constraints on the relations among all of them, including the powers of an element, which is the form in which the source's asymptotic arguments use it.

## 40.3 Obstruction Functionals

An obstruction functional is a multiplicative monotone, a function from the elements to the nonnegative reals that does not increase along the preorder and multiplies under composition. The spectral points of Chapter 8 are the instance for tensors. The basic use is the certificate theorem: if a functional takes a larger value on $x$ than on $y$, then $x$ cannot be obtained from

$y$. The theorem is immediate, and its content is in the supply of functionals. A system with few monotones has few certificates, and a system with many has many, and the existence of enough monotones to separate all non-comparable pairs is a completeness question that has a positive answer in the tensor setting only asymptotically and by duality.

## 40.4 Separation by Invariants

An invariant separates two elements if it takes different values on them. A monotone separates them in the stronger sense of ruling out a relation. The distinction between the two kinds of separation is the distinction between soundness and completeness of an obstruction method, as in Chapter 8: monotones are sound, in that they never rule out a relation that holds, and they are complete only when there are enough of them. The theory records soundness as a theorem and treats completeness as a hypothesis to be established for each structure.

## 40.5 Resource-Bounded Realization

A realization with limited expenditure is a refinement of the preorder: $x \preceq_R y$ states that $x$ can be obtained from $y$ at cost at most $R$. The general lower bound shows what a monotone says about the cost. If the monotone can increase by at most a factor $g(R)$ under a realization of cost $R$, then the cost required to go from $y$ to $x$ is at least the value of the inverse of $g$ at the ratio of the monotone's values. For the common case in which each unit of cost allows a fixed multiplicative gain, the bound is logarithmic in the ratio. This is the abstract form of the step in the source by which the value of a spectral point on the clause sum, divided by its value on a small piece, bounds the size of the circuit that could supply it.

## 40.6 Geometry Induced by Admissible Transformations

In the tensor setting the monotone, applied to the wires of a network, produces a metric on the parties, and the metric governs the cost of realizing networks. The abstract question is whether, for a general resource structure with a notion of wire, a multiplicative monotone induces a metric in the same way, and whether the geometry of the metric yields resource bounds beyond the setting in which it was discovered. This is Hypothesis 40.11 below, and it is the proposed content of the general theory that goes beyond the elementary results.

## 40.7 Curvature and Decomposability

In the source the geometry has a decisive property: the metrics are of nonpositive curvature in a sense that yields the curvature inequality for prices, and the dichotomy between the flat and contracting cases classifies how the metric can behave. The two cases give different arguments for the same conclusion, as Chapter 21 explained. The abstract theory does not

assume a curvature theorem. A nontrivial curvature theorem for a new resource structure, giving an analogue of the crossing identity and the dichotomy, would be the substantive contribution that the framework points toward, and it is not supplied here.

## 40.8 A Distinction Between Generality and Universality

The theory is general in that it applies to any structure with the stated features, and it is not universal in that it does not claim that every obstruction arises from a monotone. The elementary theorems, the certificate and the resource lower bound, hold in the generality in which they are stated. The claims that monotones suffice, that they generate geometry, and that the geometry has useful curvature are specific to structures in which they have been proved, namely tensors under restriction with the spectral points of the source. To call the theory general is to say that it has instances, and to call it universal would be to claim that the instances exhaust the field, which the monograph does not claim.

## 40.9 Formal Statements

**Definition 40.1** (Compositional Resource Structure)**.** A compositional resource structure is $\mathfrak{R} = (X, \preceq, \otimes, \mathbf{1})$, where $\preceq$ is a preorder on $X$, $(X, \otimes, \mathbf{1})$ is a commutative monoid, and $x \preceq y$, $x' \preceq y'$ imply $x \otimes x' \preceq y \otimes y'$.

**Definition 40.2** (Multiplicative Monotone)**.** A multiplicative monotone is $\varphi : X \to [0, \infty)$ with $\varphi(x) \leq \varphi(y)$ whenever $x \preceq y$, $\varphi(x \otimes y) = \varphi(x)\varphi(y)$ and $\varphi(\mathbf{1}) = 1$.

**Theorem 40.3** (Obstruction Certificate)**.** *If a multiplicative monotone $\varphi$ satisfies $\varphi(x) > \varphi(y)$, then $x \not\preceq y$. If moreover $\varphi(x)^N > \varphi(y)^M$ for some $N, M \geq 1$, then $x^{\otimes N} \not\preceq y^{\otimes M}$.*

*Proof.* If $x \preceq y$ then monotonicity gives $\varphi(x) \leq \varphi(y)$, a contradiction. For the second claim, $\varphi(x^{\otimes N}) = \varphi(x)^N$ and $\varphi(y^{\otimes M}) = \varphi(y)^M$ by multiplicativity, and the first claim applies to $x^{\otimes N}$ and $y^{\otimes M}$. □

**Definition 40.4** (Resource-Bounded Realization)**.** A resource-bounded realization is a family of relations $\preceq_R$ for $R \geq 0$ with $x \preceq_R y \Rightarrow x \preceq y$ and $x \preceq_R y$, $R \leq R'$ imply $x \preceq_{R'} y$. The minimal expenditure is $R_*(x \mid y) = \inf\{R : x \preceq_R y\}$, with $R_* = \infty$ if the set is empty.

**Theorem 40.5** (General Resource Lower Bound)**.** *Let $\varphi$ be a multiplicative monotone with $\varphi(y) > 0$, and let $g : [0, \infty) \to [1, \infty)$ satisfy $x \preceq_R y \Rightarrow \varphi(x) \leq g(R)\varphi(y)$. With $g^{-1}(t) = \inf\{R \geq 0 : g(R) \geq t\}$, $R_*(x \mid y) \geq g^{-1}(\varphi(x)/\varphi(y))$.*

*Proof.* Put $t = \varphi(x)/\varphi(y)$. If $x \preceq_R y$ then $t \leq g(R)$, so $R$ belongs to $\{R' : g(R') \geq t\}$ and $R \geq g^{-1}(t)$. Taking the infimum over all such $R$ gives $R_*(x \mid y) \geq g^{-1}(t)$, and if no $R$ exists the bound holds trivially since $R_* = \infty$. □

**Corollary 40.6** (Logarithmic Obstruction)**.** *If $g(R) = 2^{\alpha R}$ with $\alpha > 0$, then $R_*(x \mid y) \geq \frac{1}{\alpha} \log_2 \frac{\varphi(x)}{\varphi(y)}$ whenever $\varphi(x) \geq \varphi(y)$.*

*Proof.* For $t \geq 1$, $\inf\{R \geq 0 : 2^{\alpha R} \geq t\} = \frac{1}{\alpha}\log_2 t$. Apply Theorem 40.5. □

**Definition 40.7** (Cover Axiom)**.** A multiplicative monotone $\varphi$ is subadditive on covers if, whenever $x$ is covered by $x_1, \ldots, x_J$ in the structure's covering relation, $\varphi(x) \leq \sum_j \varphi(x_j)$.

**Proposition 40.8** (Covering Obstruction)**.** *If $\varphi$ is subadditive on covers, $x$ is covered by $x_1, \ldots, x_J$ and $x_j \preceq y_j$ for each $j$, then $\varphi(x) \leq J \max_j \varphi(y_j)$, that is, $\log \varphi(x) \leq \log J + \max_j \log \varphi(y_j)$ when $\varphi(y_j) > 0$.*

*Proof.* By subadditivity and monotonicity, $\varphi(x) \leq \sum_j \varphi(x_j) \leq \sum_j \varphi(y_j) \leq J \max_j \varphi(y_j)$. □

*Remark* 40.9. Proposition 40.8 is the abstract form of the step in Chapters 15 and 16 by which a circuit decomposes the tensor of the network polynomial as a sum of at most $2^k$ pieces, each a restriction of a hub network, so that the value of a spectral point on the whole is at most $2^k$ times the largest value on a hub network. The abstract theory contains this step as an instance and no more.

**Inherited Result 40.10** (Asymptotic Completeness for Tensors)**.** *For tensors under restriction, a relation between powers holds asymptotically, in the sense of Inherited Result 8.7, if and only if every spectral point satisfies the corresponding inequality. This is Theorem A.6 of the manuscript, a form of Strassen's duality, and it is the instance of completeness that the monograph uses. For an abstract structure the corresponding statement requires further hypotheses and is not assumed.*

**Hypothesis 40.11** (Geometric Obstruction Principle)**.** *For a compositional resource structure with a notion of wire $W_q(u, v)$ between labelled points, a multiplicative monotone $\varphi$ induces a function $d_\varphi$ by $\varphi(W_q(u, v)) = q^{d_\varphi(u,v)}$, and there are conditions under which $d_\varphi$ is a pseudometric whose geometric properties, such as nonpositive curvature, yield resource lower bounds beyond the arithmetic circuit setting.*

*Remark* 40.12. The abstract resource theorem is elementary and is proved above. A nontrivial curvature theorem for a new resource structure would constitute the substantive research contribution, and the monograph does not claim one. In the tensor setting the induced function is a pseudometric by Inherited Result 9.3, and its curvature properties are Inherited Results 11.3 and 12.3.

**Proof Obligation 40.13.** *Identify a resource structure other than tensors under restriction in which an induced metric has a curvature property that is both proved and yields a resource lower bound not obtainable from the elementary certificate and covering theorems alone.*

# Part XIV

# A Programme of Independent Reconstruction

# Chapter 41

# Establishing a Dependency Ledger

## 41.1 The Principal Theorem Dependencies

A mathematical result is only as secure as the results on which its proof rests, and a reader who wishes to reconstruct a result independently needs to know what those results are. The ledger of this monograph records, for each statement, its status and the statements on which its treatment depends, in the column of remarks. The present chapter turns the ledger into a formal object, a directed graph on statements with a status attached to each node, so that questions of the following kind have exact answers: which statements must be reconstructed before a given theorem can be called reconstructed, which statements are affected if a premise fails, and how far a programme of reconstruction has advanced.

For the principal separation the dependencies divide into two branches that correspond to the two alternatives of Chapter 13. The flat branch rests on the structure of the network polynomial and its realization as a permanent, on the spectral exponents and the cover axiom, on the hub realization, on the sums-with-few-terms lemma, on the midpoint convexity of the exponent, and on the facet-centres proposition, as listed in Chapter 16. The contracting branch rests on the Poincaré inequality on unitriangular groups, the existence of laws, the tower of groups and its tree decomposition, the machine constructions for the universal tape automaton, and the accelerator lemmas of Chapters 19 to 21. Both branches pass through the existence and duality of spectral points, through the curvature inequality for prices and the CAT(0) realization, and through the crossing identity. The elementary parts of the monograph, from the cut rank of graph tensors to the dominance of the contracting rate, have been proved from these inherited statements, and the chain from the principal bound to its premises is therefore a chain whose upper links are written out here and whose lower links are cited.

## 41.2 Classification of Inherited Results

A result is inherited in this monograph for one of three reasons, and the ledger distinguishes them in its remarks. Some are results of the literature, such as the completeness theorems of

Valiant and the construction of Berkowitz, which are cited under their own names and are not reproved. Some are results of the source manuscript that have not been reconstructed here, such as the curvature theorem and the Poincaré inequality, whose proofs are long and whose reconstruction is listed among the obligations. And some are statements whose printed proofs describe machine constructions and state that the programs behave as claimed only in the Lean development, so that their reconstruction would require either reproducing the programs or auditing the formal declarations. The classification matters because the three reasons call for different remedies: a citation is checked against the literature, an unreconstructed proof is reconstructed, and a machine construction is rebuilt or audited.

## 41.3 Classification of Verification Status

The status of a statement is a record of the independent evidence that has been gathered about it, and it is a different matter from the statement's truth. The formal definition below uses five values. A statement is unexamined if it appears as the source gives it and nothing has been done to it. It is statement checked if its formulation, or the formalization of its formulation, has been read against its informal meaning. It is replayed if a proof object for it has been accepted by a kernel or by a second checker. It is reconstructed if the reader has rederived it from the premises recorded in the ledger. And it is invalid if an error has been found that is not repaired. The first four values are ordered by a convention, the strength of independent evidence, and the order is a convention and not a logical implication: a replayed statement is not thereby a checked statement, since replay does not test the fidelity of the formalization, as Chapter 27 showed. In the notation of the ledger of this monograph, the entry P records a proof written in the text from stated premises, which is reconstruction on paper in the sense above, and no entry has been replayed in a formal system by the present work.

## 41.4 Recording Assumptions

Conditional results are a necessary part of the subject, and the ledger records them with the status P(c), a proof in the text that depends on a named condition such as the flat hypothesis. A conditional proof certifies an implication, and it does not certify its conclusion. The proposition on conditional certification below makes this exact: if the proof of a statement is valid relative to its premises, then the conjunction of the premises implies the statement, and the independent status of the statement is bounded by that of its premises. A reader who has checked the implication has established exactly that. The assumptions are recorded as nodes of the graph, with edges to the statements that use them, so that their consequences are visible and the consequences of their failure are computable.

## 41.5 Separating Mathematical and Computational Dependencies

Some statements of this monograph rest on finite computations, such as the exact checks of the crossing identity on small cases and the recomputation of constants. A computation is a different kind of dependency from a lemma. It has no premises in the mathematical sense, it has inputs, code, an environment and an output, and it certifies exactly the finite set of instances that it enumerates. The ledger therefore records computations as separate nodes, with edges to the statements that cite them, and a statement that rests on a computation is not reconstructed until the computation has been rerun in an environment independent of the original. The proposition on finite computations below states what a computation can and cannot establish, and the example is the classical one: a polynomial that takes prime values for forty consecutive arguments and fails at the forty-first.

## 41.6 Formal Statements

**Definition 41.1** (Dependency Graph)**.** Let $\mathcal{T}$ be a finite set of statements and $\mathcal{C}$ a finite set of computation records, disjoint from $\mathcal{T}$. A dependency graph is $G_D = (\mathcal{T} \cup \mathcal{C}, E_D)$ with $E_D \subseteq (\mathcal{T} \cup \mathcal{C}) \times \mathcal{T}$. An edge $(S, T)$ means that the recorded proof of $T$ uses $S$. Write $S \rightsquigarrow T$ if there is a directed path of length at least one from $S$ to $T$. The ledger requires that $G_D$ have no directed cycle.

**Definition 41.2** (Verification Status)**.** A verification status is a function $v : \mathcal{T} \to \{\text{unexamined}, \text{statement checked}, \text{replayed}, \text{reconstructed}, \text{invalid}\}$. On the first four values the ranking unexamined $\prec$ statement checked $\prec$ replayed $\prec$ reconstructed is adopted by convention. The value invalid is incomparable with the others.

**Definition 41.3** (Dependency Closure)**.** $\mathrm{Dep}(T) = \{S \in \mathcal{T} \cup \mathcal{C} : S \rightsquigarrow T\}$.

**Proposition 41.4** (Conditional Certification)**.** *If the proof of $T$ is valid relative to premises $S_1, \dots, S_n$, then $\bigwedge_{i=1}^{n} S_i \Rightarrow T$. If some $S_i$ remains unverified, the independent verification status of $T$ is conditional on that premise, in the sense of Definition 41.5 below.*

*Proof.* Validity of a proof relative to premises means that every step of the proof follows from the premises, earlier steps and the background logic. Assuming all $S_i$ and following the steps yields $T$, which is the implication. The second statement is the definition of the closure status. □

**Definition 41.5** (Closure Status)**.** For $T \in \mathcal{T}$ let $K(T) = (\mathrm{Dep}(T) \cap \mathcal{T}) \cup \{T\}$. The closure status is $\bar{v}(T) = \text{invalid}$ if $v(U) = \text{invalid}$ for some $U \in K(T)$, and otherwise $\bar{v}(T) = \min_{U \in K(T)} v(U)$ in the ranking of Definition 41.2.

**Proposition 41.6** (Frontier of Reconstruction)**.** *Let $G_D$ be acyclic and $T \in \mathcal{T}$. Then* $\bar{v}(T) = \text{reconstructed}$ *if and only if every statement of $K(T)$ is reconstructed. If* $\bar{v}(T) \neq$

invalid *and* $\bar{v}(T) \neq$ reconstructed*, there is* $U \in K(T)$ *that is not reconstructed and all of whose direct statement premises are reconstructed.*

*Proof.* The first claim is the definition of the minimum in a totally ordered set whose top element is reconstructed. For the second, the set $N$ of members of $K(T)$ that are not reconstructed is nonempty and finite. Since $G_D$ is acyclic, $\rightsquigarrow$ is a strict partial order, and $N$ has a minimal element $U$ for it. A direct statement premise $P$ of $U$ lies in $K(T)$, since $P \rightsquigarrow U \preceq T$, and $P \notin N$ by minimality, so $P$ is reconstructed. □

**Definition 41.7** (Reconstruction Coverage)**.** For weights $w : \mathcal{T} \to (0, \infty)$, $\mathrm{Cov}_w(G_D) = \sum_{T \in \mathcal{T}} w(T)\mathbf{1}_{\text{reconstructed}}(T) \Big/ \sum_{T \in \mathcal{T}} w(T)$.

**Proposition 41.8** (Coverage Is a Progress Indicator)**.** $\mathrm{Cov}_w \in [0, 1]$*, and* $\mathrm{Cov}_w = 1$ *if and only if every statement is reconstructed. Raising the status of one statement to reconstructed strictly increases* $\mathrm{Cov}_w$*. If at least one statement is reconstructed and at least one is not, then every value in* $(0, 1)$ *is the coverage of the same status function for a suitable choice of weights.*

*Proof.* The numerator is at most the denominator, and equal exactly when the indicator is one for every statement, since the weights are positive. Raising one status adds a positive weight to the numerator and leaves the denominator unchanged. For the last claim let $W_R$ and $W_U$ be the total weights of the reconstructed and the other statements. Then $\mathrm{Cov}_w = W_R/(W_R + W_U)$, and the ratio $W_R/W_U$ can be given any positive value by scaling the weights within each class, so every value in $(0, 1)$ occurs. □

*Remark* 41.9. The last statement of Proposition 41.8 is the reason that coverage is a progress indicator and not a probability that the principal theorem is correct. It depends on a weighting that is a choice of the maintainer, it concerns status and not truth, and a reconstruction can itself contain an error. The unweighted coverage of a ledger is also not a measure of closeness to a proof of the principal theorem, since the closure status of Definition 41.5 is determined by the closure of that theorem alone, and statements outside the closure do not contribute to it.

**Proposition 41.10** (Finite Computations Certify Their Instances)**.** *Let* $f(n) = n^2 + n + 41$*. A computation that tests primality of* $f(n)$ *for* $n = 0, 1, \ldots, 39$ *certifies each of these forty statements, and no statement about any other* $n$*. In particular it does not certify the universal statement that* $f(n)$ *is prime for every* $n \geq 0$*, which is false since* $f(40) = 41^2$*.*

*Proof.* The computation tests the forty listed instances and records its outputs. An instance $n = 40$ has $f(40) = 1600 + 40 + 41 = 1681 = 41^2$, which is composite, so the universal statement has a counterexample outside the tested range. A statement certified for a finite set of arguments is therefore not implied for any argument outside it. □

**Proof Obligation 41.11.** *The ledger of this monograph records dependencies in its remarks column in prose. Export it as a graph in the sense of Definition 41.1, with a node for each*

*statement and computation, with typed edges, and with the status function of Definition 41.2, and compute closure statuses and coverage for the principal separation.*

# Chapter 42

# Elementary Worked Examples

## 42.1 A Two-Party Wire Tensor

The simplest tensor with a nontrivial flattening is the wire of dimension two between two parties, $W_2 = e_0 \otimes e_0 + e_1 \otimes e_1$. Its flattening, with the rows indexed by the state of the first party and the columns by the state of the second, is the identity matrix of size two. The rank of the flattening is therefore two, and the wire cannot be written as a single product of a tensor on the first party and a tensor on the second. This is the unit of all the later counts: every crossing wire of dimension $q$ contributes a factor $q$ to the rank across a cut that it crosses, and a wire that stays on one side contributes nothing.

## 42.2 A Path Network

A path on $k$ nodes with a wire of dimension $q$ on each edge has, across any cut that separates an initial segment from the remaining nodes, exactly one crossing edge, so the rank across it is $q$. With $q = 2$ and a balanced cut of a path on eight nodes the rank is 2. The rank is the same for every $k$, and in particular it does not grow with the number of nodes. This is what makes the path network cheap in Chapter 14: a computation that propagates through the path needs only a state of size $q$ to be handed from one segment to the next. The exact computation of the flattening of the path tensor for $k = 4, 6, 8$ with $q = 2$, carried out in Chapter 43, returns the rank 2 in each case.

## 42.3 A Complete Network

In the complete graph on $k$ nodes with the same wires, a cut with $s$ nodes on one side crosses $s(k - s)$ edges, and the rank is $q^{s(k-s)}$. At $q = 2$, $k = 8$ and $s = 4$ the exponent is sixteen, and the rank is $2^{16} = 65\,536$, in contrast with the value 2 for the path on the same number of nodes. The ratio between the two is the whole content of the contrast between a locally coupled and a globally coupled system in this monograph. The value for $k = 8$ is the formula of Proposition 9.6. Direct computation of the flattening was carried out for the

complete graphs on four and five nodes, where it returns ranks 16 and 64 for cuts with two nodes on one side, and for cuts with a single node on one side in the complete graphs on five and six nodes, where it returns 16 and 32. These agree with the formula in every case.

## 42.4 Crossing Wires

The cost of a cut depends on which pairs of parties are joined, and not only on how many wires there are. Let four parties be labelled $1, 2, 3, 4$ and consider two networks with two wires of dimension $q$ each: the first joins 1 to 3 and 2 to 4, the second joins 1 to 4 and 2 to 3. Across the cut $\{1, 2\} \mid \{3, 4\}$ both networks have rank $q^2$. Across the cut $\{1, 3\} \mid \{2, 4\}$ the first network has rank 1, since neither wire crosses, while the second has rank $q^2$, since both do. The two networks are the same up to a relabelling of the parties 3 and 4, and they are incompatible in the sense that a cut that costs nothing in one costs the maximum in the other. This is the elementary form of the phenomenon that the crossing identity of the source quantifies, namely that decompositions that are cheap along one partition are expensive along another. The statement here is exact and elementary, and it does not rely on the crossing identity, which remains an inherited result.

## 42.5 Determinant and Permanent in Characteristic Two

For a two by two matrix with entries $a, b, c, d$ the permanent is $ad + bc$ and the determinant is $ad - bc$. They differ by $2bc$, so they are different polynomials in every characteristic except two, and in characteristic two they coincide. The same holds in every size: the permanent and the determinant differ by twice the sum of the products over odd permutations, and the difference vanishes exactly when the characteristic is two. Exact symbolic computation confirms the case of size three, where the difference has three terms with coefficient two, and a computation over the two-element field on two hundred random zero-one matrices of size four finds no case in which the permanent and determinant differ. The coincidence is the reason that the statement of the reported separation excludes characteristic two and that determinantal transfer of Chapter 23 applies only outside it.

## 42.6 Boolean Equivalence and Lost Polynomial Structure

The polynomials $f(x) = x^2 - x$ and $g(x) = 0$ take the same value at both points of the Boolean cube, so $E(f) = E(g)$, and they are different polynomials. The extraction of the part of degree one in the variable $x$ gives $-x$ for $f$ and 0 for $g$, so the extraction takes different values on the two members of one fiber. The circuit sizes differ as well: the list consisting of the constant 0 computes $g$ and has length one, so the cheapest size in the class of $f$ under the Boolean quotient is one, while $f$ itself requires at least three entries, since a list of length at most two computes only a variable, a constant, or a sum or product of its first entry with itself, and none of these is $x^2 - x$ in any characteristic. Here the quotient

cost and the formal cost already disagree for a single variable. Chapter 38 shows that the discrepancy is unbounded in general.

## 42.7 Verification and Discovery Costs

If a certificate costs $D$ to discover and $a$ per use to apply, the average cost over $N$ uses is $a + D/N$. For $D = 1000$ and $a = 1$ the averages for $N = 1, 10, 100, 1000$ and $10^6$ uses are 1001, 101, 11, 2 and 1.001. The limit is $a$, and the original expense is negligible per use after enough uses. The break-even of Chapter 26 gives a sharper reading. If rediscovery costs $d = 11$ per use, reuse costs $a = 1$ per use, the establishment cost is $F = 1000$ and maintenance is 4 per use, then reuse is cheaper than rediscovery exactly when $N(11 - 1 - 4) > 1000$, that is for $N \geq 167$, since $166 \cdot 6 = 996$ falls short and $167 \cdot 6 = 1002$ does not. If the maintenance rate were 10 per use, the saving per use would vanish and no number of uses would suffice.

## 42.8 Formal Statements

**Proposition 42.1** (Wire Rank)**.** *The flattening of $W_2$ is the identity matrix of size two, and $r(W_2) = 2$. More generally the flattening of $W_q(u, v)$ is the identity of size $q$.*

*Proof.* The entry of the flattening at row $i$ and column $j$ is the coefficient of $e_i \otimes e_j$ in $W_q$, which is $\delta_{ij}$. The identity of size $q$ has rank $q$. □

**Proposition 42.2** (Path Versus Complete Network)**.** *For the path on $k$ nodes with every wire of dimension $q$, every flattening rank across an initial segment is $q$. For the complete graph on $k$ nodes with every wire of dimension $q$, the rank across a set of $s$ nodes is $q^{s(k-s)}$. For $q = 2$, $k = 8$ and $s = 4$ these are 2 and 65 536.*

*Proof.* By Proposition 9.6 the rank is the product of the dimensions of the crossing wires. An initial segment of a path is joined to its complement by one edge. A set of $s$ nodes in the complete graph is joined to its complement by $s(k - s)$ edges, one for each pair with one node on each side. Evaluating with $q = 2$ gives 2 and $2^{16} = 65\,536$. □

**Proposition 42.3** (Crossing Wires)**.** *Let $N_1 = W_q(1, 3)W_q(2, 4)$ and $N_2 = W_q(1, 4)W_q(2, 3)$ on four parties. Their flattening ranks are $r_{\{1,2\}}(N_1) = r_{\{1,2\}}(N_2) = q^2$, $r_{\{1,3\}}(N_1) = 1$ and $r_{\{1,3\}}(N_2) = q^2$.*

*Proof.* By Proposition 9.6, each rank is $q$ raised to the number of wires with exactly one endpoint in the set. For $\{1, 2\}$ both wires of each network cross. For $\{1, 3\}$ the wires $\{1, 3\}$ and $\{2, 4\}$ of $N_1$ lie within a side, and the wires $\{1, 4\}$ and $\{2, 3\}$ of $N_2$ both cross. □

**Proposition 42.4** (Permanent and Determinant Differ by Twice the Odd Terms)**.** *For an $n \times n$ matrix of indeterminates,* $\operatorname{perm} - \det = 2 \sum_{\sigma \text{ odd}} \prod_i x_{i\sigma(i)}$. *For $n \geq 2$ this polynomial is nonzero in every characteristic other than two, and it is zero in characteristic two. In particular* $\operatorname{perm} \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = 2bc$.

*Proof.* Since $\det = \sum_\sigma \operatorname{sgn}(\sigma) \prod_i x_{i\sigma(i)}$ and $\operatorname{perm} = \sum_\sigma \prod_i x_{i\sigma(i)}$, the difference has coefficient $1 - \operatorname{sgn}(\sigma)$ at the monomial of $\sigma$, which is 0 for even and 2 for odd permutations. Distinct permutations give distinct monomials, so the polynomial is nonzero exactly when $2 \neq 0$ in the field and when an odd permutation exists, which requires $n \geq 2$. For $n = 2$ the only odd permutation is the transposition, giving $2bc$. □

**Proposition 42.5** (Boolean Equivalence of a Single Variable)**.** *For $f = x^2 - x$ and $g = 0$, $E(f) = E(g)$ and $f \neq g$. The extraction of the part of degree one in $x$ takes the value $-x$ on $f$ and $0$ on $g$. The least length of a list computing $g$ is one, and the least length of a list computing $f$ is at least three.*

*Proof.* Both polynomials vanish at $x = 0$ and $x = 1$, and they differ as polynomials. The part of degree one of $x^2 - x$ is $-x$. The list consisting of the constant 0 computes $g$, and no list is empty. A list of length one is a variable or a constant. A list of length two has a first entry that is a variable or a constant, and a second entry that is a variable, a constant, or a sum or product of the first entry with itself, giving a polynomial of the form $x$, $2x$, $x^2$, or a constant. The polynomial $x^2 - x$ has two monomials, each with a coefficient that is nonzero in every characteristic, so it is none of these. □

**Proposition 42.6** (Amortized Average Cost)**.** *For $D \geq 0$ and $a \geq 0$, $\overline{C}_N = a + D/N$ is nonincreasing in $N$, strictly decreasing if $D > 0$, and tends to $a$. For $D = 1000$ and $a = 1$ the values at $N = 1, 10, 100, 1000, 10^6$ are $1001, 101, 11, 2, 1.001$. In the model of Proposition 26.2 with $d = 11$, $a = 1$, $F = 1000$ and $M(N) = 4N$, reuse is cheaper than rediscovery exactly for $N \geq 167$.*

*Proof.* The map $N \mapsto D/N$ is nonincreasing and tends to zero. The listed values are direct evaluation. By Proposition 26.2 reuse is cheaper iff $N(d - a - \mu) > F$ with $d - a - \mu = 6$, so $6N > 1000$, which holds iff $N \geq 167$, since $6 \cdot 166 = 996$ and $6 \cdot 167 = 1002$. □

# Chapter 43

# Computational Experiments and Reproducibility

## 43.1 Independent Implementations

A computation reported in a mathematical text is a claim about the output of a program, and the claim is only as good as the program and the environment that produced it. Two kinds of independence are useful. An independent implementation computes the same quantity by a different method, so that a bug in one is unlikely to appear in the other. An independent environment runs the same program on different software and hardware, so that a dependence on the particular library is detected. The experiments of this chapter use the first kind throughout: the ranks of flattenings are computed once by floating-point linear algebra and once by exact elimination modulo a large prime, and the two agree in every case reported. The second kind has not been carried out, and the reproducibility record below states the environment that was used.

## 43.2 Exact Arithmetic

An identity between tensors or polynomials is exact, and its verification should be exact. For a claimed identity $T = \sum_{j} T_j$ the residual $R = T - \sum_{j} T_j$ is computed in exact arithmetic, over the rationals or a finite field, and the check succeeds when $R = 0$. The crossing-identity checks of Chapter 11 are of this kind. They compute the relevant matrices over the polynomial ring in the indeterminate with exact coefficients, they test the degree bound, the reflection symmetry, the vanishing outside the support conditions and the number and values of the nonzero entries, and they succeed in the cases examined, which are small. The same chapter records that the rank-bound part of the check does not discriminate, since the bound holds with room to spare, and the exact computations are evidence for the identity in the cases computed and nothing more, by Proposition 41.10.

## 43.3 Numerical Geometric Experiments

The curvature inequality for prices is the statement that a certain margin is nonnegative, and the margin can be computed for any metric space with midpoints. It is zero in every Euclidean space, by the parallelogram law, nonnegative in every space of nonpositive curvature, and negative in some configurations in spaces of positive curvature. The experiments computed the margin in three spaces: Euclidean three-space, a metric tree with three branches, and the two-sphere with the great-circle metric. In Euclidean space over $10^5$ random configurations the margin ranged over an interval of width below $4 \times 10^{-14}$ about zero, which is rounding error. In the tree, over $2 \times 10^5$ random configurations, the smallest margin was about $-5 \times 10^{-15}$, again rounding error, consistent with nonnegativity. On the sphere, over about $10^5$ random configurations with the pair of endpoints not nearly antipodal, the margin was negative in every case, with a smallest value of about $-8.6$ over the sample. The sphere configurations show that the experiment detects positive curvature, so that the numerical agreement in the other two spaces is not vacuous. The results say nothing about the space of prices of the source, for which nonpositive curvature is an inherited theorem and not an experimental finding.

## 43.4 Parameter Sensitivity

Asymptotic claims are about the behavior of a function as its argument tends to infinity, and finite data cannot decide them. To see how weak the constraint is, take data generated by $\log L(m) = (\log m)^{4/3}$ at 121 equally spaced values of $\log m$ between 4 and 64, and fit the model $c(\log m)^{\alpha}$ by the best constant in the logarithmic sense. For $\alpha = 4/3$ the fit is exact. For $\alpha = 5/4$ the best constant is about 1.32 and the largest discrepancy between the logarithms of the fitted values and the data is about 0.16, a factor of less than 1.2. For $\alpha = 7/5$ the best constant is about 0.80 and the largest discrepancy is about 0.13. A discrepancy of this size is smaller than the variation of most constant factors in a measurement, so data of this range cannot separate nearby exponents, and the diagnostic quantity of Definition 43.6 can suggest an exponent but cannot establish one. The data in this experiment are synthetic and are generated from the formula they are fitted with, and the experiment bears on the resolving power of the diagnostic and not on any actual circuit family.

## 43.5 Formalization Strategy

A computational check and a formal proof play different roles. A check establishes finitely many instances, a formal proof establishes a universal statement relative to a kernel and a formalization. The two meet at the definition of the statement, and the formalization of Chapter 27 shows what can go wrong there. For the flat-case argument reconstructed in Chapters 14 to 16, a formalization would proceed from the elementary parts outward: the cut rank of graph tensors, which rests on the multiplicativity of matrix rank, then the

balanced decomposition and the Euclidean estimates, and last the parts that depend on the inherited spectral results, which would enter as hypotheses of the formal theorems, so that the formal statements record their conditional character exactly as the ledger does. A formalization that proves the flat-case rate from hypotheses stating the inherited results would be a replay of the conditional parts and would leave the inherited parts at their present status. This ordering is a plan and none of it has been carried out here.

## 43.6 Independent Verification Records

A reproducibility record fixes what a reader needs to rerun a computation: the hash of the source, the environment, the parameters, the commands and the outputs. For the computations of this chapter the source of the checks of Chapter 11 is the manuscript whose file has the SHA-256 digest `882f5eefb1e067cc8c99ddfe660c10d1d67f9fcee7b69364bd0bab06c0c1f97c`. The environment was Python 3.13 with SymPy 1.14.0 and NumPy 2.5.3. The parameters are those stated in the sections above: the rank computations for the path graphs on four, six and eight nodes and for complete graphs on four and five nodes with wire dimension two, the margins in three spaces with the sample sizes stated, and the synthetic scaling fit. The ranks were computed twice, in floating point and by exact elimination modulo $2^{31} - 1$. The records state what was run, and the record of a second environment is an open item.

## 43.7 Formal Statements

**Definition 43.1** (Exact Identity Residual)**.** For a claimed identity $T = \sum_j T_j$ between tensors over a field or ring, the residual is $R = T - \sum_j T_j$. An exact computational check succeeds when $R = 0$ in exact arithmetic.

**Definition 43.2** (Numerical Residual)**.** For a floating-point computation, $\epsilon_{\mathrm{num}} = \|R\| / \max\{1, \|T\|\}$.

**Proposition 43.3** (Small Numerical Residual Does Not Establish an Identity)**.** *For every $\delta \in (0, 1)$ there are tensors $T, T_1$ with $T \neq T_1$ and $\epsilon_{\mathrm{num}} < \delta$ for the claimed identity $T = T_1$.*

*Proof.* Take a tensor $T_1$ of norm $\frac{1}{2}$, a tensor $u$ of norm one, and $T = T_1 + \frac{\delta}{2} u$, so that $\|T\| \leq 1$. Then $R = T - T_1 = \frac{\delta}{2} u \neq 0$ and $\epsilon_{\mathrm{num}} = \|R\| / \max\{1, \|T\|\} = \delta/2 < \delta$. □

**Definition 43.4** (Geometric Inequality Margin)**.** For a metric space $(X, d)$, points $x, x', z \in X$ and a midpoint $y$ of $x$ and $x'$, the margin is $M(x, x', y, z) = \frac{1}{2} d(z, x)^2 + \frac{1}{2} d(z, x')^2 - \frac{1}{4} d(x, x')^2 - d(z, y)^2$. The curvature inequality of Inherited Result 12.3 in its classical form states $M \geq 0$.

**Proposition 43.5** (Margin in Euclidean and Spherical Cases)**.** *In a Euclidean space with $y = \frac{1}{2}(x + x')$, $M = 0$ for every $z$. On the unit two-sphere with the great-circle metric, for $x = (1, 0, 0)$, $x' = (0, 1, 0)$, $y = (1, 1, 0)/\sqrt{2}$ and $z = (0, 0, 1)$, $M = -\pi^2/16 < 0$.*

*Proof.* In Euclidean space the parallelogram law gives $\|z-x\|^2+\|z-x'\|^2 = 2\|z-y\|^2+\frac{1}{2}\|x-x'\|^2$, which rearranges to $M = 0$. On the sphere $d(x, x') = \pi/2$, $y$ is the midpoint of the arc, and $z$ is at distance $\pi/2$ from each of $x, x', y$. Then $M = \frac{1}{2}\cdot\frac{\pi^2}{4}+\frac{1}{2}\cdot\frac{\pi^2}{4}-\frac{1}{4}\cdot\frac{\pi^2}{4}-\frac{\pi^2}{4} = -\frac{\pi^2}{16}$. □

**Definition 43.6** (Empirical Scaling Diagnostic)**.** For quantities $L(m)$, the diagnostic is $Q_\alpha(m) = \log L(m)/(\log m)^\alpha$.

**Proposition 43.7** (Finite Data Do Not Determine the Exponent)**.** *For every finite set of values $m_1, \dots, m_s$, every positive values $\lambda_1, \dots, \lambda_s$ and every pair of exponents $\alpha \neq \alpha'$, there are functions $L$ and $L'$ with $L(m_i) = L'(m_i) = \lambda_i$ for all $i$ and with $\log L(m) \sim (\log m)^\alpha$ and $\log L'(m) \sim (\log m)^{\alpha'}$ as $m \to \infty$.*

*Proof.* Let $M = \max_i m_i$. Set $L(m) = \lambda_i$ at $m = m_i$, and for $m \notin \{m_i\}$ define $\log L(m) = (\log m)^\alpha$ for $m > M$ and any positive value for the remaining $m$. Define $L'$ in the same way with $\alpha'$. The two agree on the finite set and have different asymptotics. □

**Definition 43.8** (Reproducibility Record)**.**

$$\mathcal{R}_{\text{exp}} = (\text{source hash}, \text{environment}, \text{parameters}, \text{commands}, \text{outputs}).$$

**Proposition 43.9** (Path and Complete-Graph Ranks by Direct Computation)**.** *For wire dimension two, the flattening of the graph tensor of the path on $k = 4, 6, 8$ nodes across the first $k/2$ nodes has rank 2 in each case. The flattening of the graph tensor of the complete graph on four nodes across two nodes has rank 16 and on five nodes across two nodes has rank 64, and across a single node has rank 16 on five nodes and 32 on six nodes. For the networks $N_1, N_2$ of Proposition 42.3 with $q = 2$ the ranks across $\{1, 2\}$ are $4, 4$ and across $\{1, 3\}$ are $1, 4$. Each agrees with the formula of Proposition 9.6.*

*Proof.* These are results of exact computation, in the environment of the record above, by two independent rank computations that agree. The statement is of finite instances, by Proposition 41.10, and is consistent with Proposition 9.6, which proves the general formula. □

**Proof Obligation 43.10.** *Rerun the computations of this chapter and of the crossing-identity checks of Chapter 11 in an environment independent of the original, with a second implementation of exact rank, and record the result in the form of Definition 43.8.*

# Chapter 44

# Maintenance, Correction, and Publication

## 44.1 A Maintained Mathematical Repository

A result that is published once and not maintained decays in a specific way: errors that are found are not recorded against it, definitions drift from their uses, and the dependencies on which it rests change beneath it. A maintained repository keeps the artifact together with its dependency record and its reproducibility record, and it applies to it a policy of versioning. The artifact of Chapter 30 is the tuple of statements, proofs, dependencies and records, and a version of it at each time. The present chapter fixes the notation for versions and for the propagation of a correction through the dependency graph of Chapter 41, so that the effect of a correction on the status of other results is a computation and not an impression.

## 44.2 Versioned Intellectual Responsibility

Each revision is made by someone, and the record of who made which revision against which objection is part of the artifact. Chapter 30 showed that the record of corrections is monotone, that it grows by exactly the objections incorporated and the maintainers who incorporated them. The present chapter adds the observation that this record attaches responsibility to specific revisions and not to the artifact as a whole: a maintainer who incorporates a correction is accountable for that revision, and the accountability does not extend to results that the revision did not touch. The versioned form makes this precise, since each revision has a domain, the set of statements it changes, and a maintainer is accountable for the revisions through which the record shows an action, with the dependency graph indicating which other statements are affected without being changed.

## 44.3 Corrections Without Concealment

A correction that removes the trace of the error is a different act from one that records it. The monotone record of Chapter 30 requires the latter: a revision adds to the record and removes nothing from it. For a mathematical text this means that a retracted lemma remains in the version history with its status marked invalid, together with the statements whose closure status changed because of it, and that a reader of the current version can see both the present state and the way it was reached. The reason is practical as much as ethical. A reader who reconstructs a result from an earlier version needs to know that the earlier version contained an error and where, and a repository that conceals the error leaves such readers to rediscover it.

## 44.4 Contributions and Criticism

An objection to a result is an input to its maintenance. The objection may point to an error, to a gap, to a missing hypothesis, or to an ambiguity, and each requires a different revision. The distinction between an objection that is incorporated and one that is rejected is recorded, with the reason for rejection, since a rejected objection that was correct is a latent error and a rejected objection that was incorrect is evidence of what the text leaves unclear. A contribution in the sense of Chapter 30 may be an objection that was incorporated, a repair, a new proof of an existing statement, or a new statement, and the authorship relation requires the contribution together with an identifiable relationship to the accountability of the artifact. An objection raised without contribution to the repair creates no authorship, and an objection that triggers a correction is recorded in the correction record without thereby making its author a maintainer.

## 44.5 The Distinction Between Maintenance and Authority

Maintenance is the capacity and the willingness to keep an artifact correct, and authority is the standing to have one's claims accepted. The two can come apart. A maintainer who keeps a result correct has no authority beyond what the result's proofs give it, and a person with institutional authority who does not maintain a result is not thereby a maintainer. The mathematical content of this distinction is the independence that Chapter 30 proved between responsibility and validity. A result is valid or not by its proofs and their premises, and the status ledger records the evidence for validity, whoever maintains it. The institutional question of who may modify an artifact is a question about rights, and the next section records the rights that independent maintenance requires.

## 44.6 Authorship in Practice

For a repository in which several people contribute, the practical record of authorship is a log of revisions with attributions and a statement of the rights each contributor holds. The normative relation of Chapter 30 can be read off such a log: a contributor with documented contributions to a statement and an identifiable accountability for it is an author of that statement in the prospective sense. A repository that credits contributors who have no such accountability, or omits those who have, produces a record that does not satisfy the relation. The monograph proposes the relation and does not claim that any actual repository satisfies it, and the choice of a policy for a specific repository is a matter for its maintainers.

## 44.7 Formal Statements

**Definition 44.1** (Versioned Artifact)**.** A versioned artifact is a sequence $W^{(n)} = (T^{(n)}, \Pi^{(n)}, D^{(n)}, R^{(n)})$ of artifacts in the sense of Chapter 30, with statements $T^{(n)}$, proofs $\Pi^{(n)}$, dependency record $D^{(n)}$ and reproducibility and correction record $R^{(n)}$.

**Definition 44.2** (Revision History)**.** A revision history is a chain $W^{(0)} \xrightarrow{\delta_1} W^{(1)} \xrightarrow{\delta_2} \cdots \xrightarrow{\delta_n} W^{(n)}$ in which each $\delta_t$ is a revision applied at time $t$ by a maintainer with the capacity to apply it, and the record satisfies $R^{(0)} \subseteq R^{(1)} \subseteq \cdots \subseteq R^{(n)}$, as in Proposition 30.4.

**Definition 44.3** (Affected Set)**.** For a dependency graph $G_D$ and a statement $S$, $\mathrm{Aff}(S) = \{T \in \mathcal{T} : S \rightsquigarrow T\}$.

**Proposition 44.4** (Dependency Invalidation)**.** *Let* $\mathrm{Inv} = \{S \in \mathcal{T} : v(S) = \text{invalid}\}$*. Then* $\{T : \bar{v}(T) = \text{invalid}\} = \mathrm{Inv} \cup \bigcup_{S \in \mathrm{Inv}} \mathrm{Aff}(S)$*. Hence every statement whose recorded proof uses an invalid statement, directly or through intermediate statements, has closure status invalid and must be reconsidered. This does not imply that any such statement is false.*

*Proof.* By Definition 41.5, $\bar{v}(T) = \text{invalid}$ iff some $U \in K(T)$ is invalid, that is, iff $T \in \mathrm{Inv}$ or some $S \in \mathrm{Inv}$ satisfies $S \rightsquigarrow T$, which is $T \in \mathrm{Aff}(S)$. The last sentence holds because the closure status records the status of the recorded proof and its premises, and a statement can be true while its recorded proof is not valid. □

**Definition 44.5** (Alternative Proofs and Support)**.** Assume each statement $T$ has a finite nonempty family $\Pi(T)$ of proofs, and each proof $p$ has a set $\mathrm{pre}(p) \subseteq \mathcal{T}$ of premises. The dependency graph has an edge $(U, T)$ whenever $U \in \mathrm{pre}(p)$ for some $p \in \Pi(T)$. Given a set $B \subseteq \mathcal{T}$ of blocked statements, the supported set $\mathrm{Sup}(B)$ is the least set such that $T \in \mathrm{Sup}(B)$ whenever $T \notin B$ and some $p \in \Pi(T)$ has $\mathrm{pre}(p) \subseteq \mathrm{Sup}(B)$.

**Proposition 44.6** (Invalidation Is an Upper Bound on Loss of Support)**.** *For a statement* $S$*,* $\mathrm{Sup}(\varnothing) \setminus \mathrm{Sup}(\{S\}) \subseteq \{S\} \cup \mathrm{Aff}(S)$*, and the inclusion can be strict.*

*Proof.* Let $T \in \mathrm{Sup}(\varnothing)$ be first derived at stage $k$ through a proof $p$ with premises derived at earlier stages, and suppose $T \notin \mathrm{Sup}(\{S\})$. Induct on $k$. If $T = S$ the claim holds. Otherwise

some premise $P$ of $p$ is not in $\mathrm{Sup}(\{S\})$, for if all were, then $T$ would be derived there. By the induction hypothesis $P \in \{S\} \cup \mathrm{Aff}(S)$, and since $(P, T)$ is an edge, $T \in \mathrm{Aff}(S)$. For strictness let $\mathcal{T} = \{S, S', T\}$ with $\Pi(S)$ and $\Pi(S')$ each a single proof with no premises, and $\Pi(T)$ consisting of one proof with premise $S$ and one with premise $S'$. Then $T \in \mathrm{Aff}(S)$ but $T \in \mathrm{Sup}(\{S\})$ through $S'$. □

**Definition 44.7** (Independent Maintenance)**.** Let $\mathrm{Avail}(a)$ be the resources available to a researcher $a$. An artifact $W$ is independently maintainable relative to $a$ if its source materials, its dependencies, the verification mechanisms needed to check its claims, and the rights to modify it all lie in $\mathrm{Avail}(a)$, without requiring continued access to the original discovery process.

**Proposition 44.8** (Maintainability Is Monotone in Availability and Needs Rights)**.** *If* $\mathrm{Avail}(a) \subseteq \mathrm{Avail}(a')$ *and* $W$ *is independently maintainable relative to* $a$*, then it is independently maintainable relative to* $a'$*. If the rights to modify* $W$ *are not in* $\mathrm{Avail}(a)$*, then* $W$ *is not independently maintainable relative to* $a$*, however complete its sources, dependencies and verification mechanisms. If the sources, dependencies and verification mechanisms of* $W$ *lie in* $\mathrm{Avail}(a)$ *and copying with modification is permitted, then a copy* $W'$ *of* $W$ *is independently maintainable relative to* $a$*, and revisions to* $W'$ *do not enter the correction record of* $W$*.*

*Proof.* The first claim is the monotonicity of set inclusion applied to the four required components. The second is the definition, which requires all four components. For the third, the copy has the same sources and dependencies and verification mechanisms, and the permission to modify supplies the fourth component for the copy. A revision to $W'$ is a revision of a different artifact, so by Definition 44.2 it enters the record of $W'$ and not the record of $W$. □

**Proof Obligation 44.9.** *Specify a concrete repository layout that records, for the artifact of this monograph, the statements, proofs, dependency graph of Obligation 41.11, reproducibility records and revision history, and a policy for objections and corrections, and apply it to the principal separation.*

# Part XV

# Conclusion: Mathematics After Discovery

# Chapter 45

# The Transformation of Difficulty

## 45.1 From Search to Verification

The monograph began with the observation that the difficulty of a mathematical proposition is not a fixed property of it. Before a certificate exists, the task faced by a reader is a search, over a space of candidate proofs that may be enormous. After a certificate exists and remains accessible, the task is the evaluation of a verification relation on the pair of the statement and the certificate, and for the usual proof systems this evaluation takes time polynomial in the length of the pair. The transformation is real, and it is conditional on three things that the preceding parts separated: that a certificate has been found, that it remains available to the party who needs it, and that the cost of keeping it available does not outgrow the saving it provides. The synthesis of this chapter states these conditions in a single theorem and records what the theorem does not say.

## 45.2 Reuse and the Disappearance of the Original Search

When a certificate is used repeatedly, the expense of its discovery is paid once and the expense of its verification and application is paid at each use. Averaged over the uses, the discovery cost contributes a term that decays like the reciprocal of the number of uses, and it vanishes in the limit if the maintenance cost grows more slowly than the number of uses. This is the accounting of Chapters 2 and 26 in its asymptotic form. The statement concerns an average over many uses of a single completed construction, and it explains in what sense the result of a hard search becomes cheap to those who inherit it. It does not say that the search was easy, and it does not say that the next search will be.

## 45.3 What the Transformation Does Not Show

The principle that completed and preserved constructions eliminate the necessity of repeated search is not the claim that all of mathematics becomes polynomial time. The set of provable

statements is a language that can be undecidable, as Chapter 1 showed, and a polynomial-time verifier does not place that language in the class of polynomial-time decidable languages. The transformation applies to a statement once a certificate is held, and it is silent about statements for which none is held. Nor does it address accessibility beyond the cost model: a certificate that exists and is not accessible has an infinite reconstruction cost in the sense of Chapter 1, and the reduction of difficulty is then unavailable to a party in that environment, however low the verification cost would have been.

## 45.4 Formal Statements

**Theorem 45.1** (Certificate-Conditioned Reuse)**.** *Let $V$ be a verification relation computable in time polynomial in $|T| + |\pi|$, and let $\pi$ be a valid certificate of $T$. First, the language $L_V = \{(T, \pi) : V(T, \pi) = 1\}$ belongs to* P*, and the set of provable statements $L_{\exists V}$ need not be decidable. Second, if $\pi$ is accessible in the environment $K$, each use costs the verification and application expenditure and does not include the discovery cost $C_D(T)$. Third, suppose that in the cumulative expenditure of Definition 26.1 the quantities $C_D, C_V, C_I$ are finite constants, that $C_M(N)/N \to 0$, and that the mean application expenditure $\frac{1}{N}\sum_{i \le N} A_i$ has limit superior $\bar{a} < \infty$. Then $\limsup_{N\to\infty} C(N)/N = \bar{a}$, with an ordinary limit whenever the mean application expenditure converges, and the contribution of the discovery cost to the average, $C_D/N$, tends to zero. If $\pi$ is not accessible, so that $C_K(T) = \infty$, none of the conclusions about reuse applies.*

*Proof.* The first part is Proposition 1.5. For the second, the cost of confirming $T$ with an accessible certificate is $C_K(T) = \inf\{C_{\text{access}}(\pi) + C_V(T, \pi)\}$ over accessible valid certificates, which contains no term for the search. For the third, write $C(N)/N = (C_D + C_V + C_I + C_M(N))/N + \frac{1}{N}\sum_{i \le N} A_i$. The first term tends to zero because the constants are divided by $N$ and $C_M(N)/N \to 0$. Hence the limit superior of $C(N)/N$ equals $\bar{a}$, and $C(N)/N$ converges whenever the mean application expenditure does. The share of the discovery cost is $C_D/N \to 0$. If $C_K(T) = \infty$, Proposition 1.9 gives that existence of a certificate does not bound the cost of use, and the accounting of reuse does not apply. □

*Remark* 45.2. Theorem 45.1 combines Chapters 1, 2 and 26. The break-even condition of Proposition 26.2 is its finite-horizon form, and Corollary 26.3 is its failure under restricted access. The theorem is a statement about an accounting model whose parameters, in any actual setting, require evidence that the monograph does not supply (Obligation 26.11).

# Chapter 46

# Geometry as an Explanatory Language

## 46.1 A Common Pattern of Argument

The lower-bound arguments of Parts V to VII have a common form. A circuit of a given size is shown to induce a decomposition of a tensor into pieces of a restricted kind. A spectral functional, which is monotone under restriction and multiplicative under products, is evaluated on both sides of the decomposition, and the geometry of the induced pseudometric controls the value on the pieces. If the value on the whole exceeds what the pieces can supply, by a quantitative gap, the size of the circuit is bounded below. The pattern has three ingredients, a decomposition, an invariant and a geometric gap, and the content of any particular application is the independent establishment of the three.

## 46.2 What the Pattern Explains

The pattern explains why the complete network is hard and the path network is easy in a way that the bare cut-rank computation does not. The cut rank records the representational cost across one partition. The spectral value records the cost across all partitions simultaneously, weighted by the geometry, and the geometry makes the incompatibility of different partitions quantitative. The flat and contracting alternatives then classify how the geometry can behave, and each yields a gap by its own mechanism. In this sense the geometry is an explanatory language: it gives names to the quantities that must be compared and it supplies tools, the curvature inequality and the CAT(0) realization, for comparing them.

## 46.3 Limits of the Language

An explanatory language is useful to the extent that it applies beyond the case that produced it. The generality of Chapter 40 is the generality of the template, and the elementary

theorems hold for any structure with the stated features. The substantive ingredients, a curvature theorem for a new resource structure and a decomposition lemma for a new family of circuits, are not supplied by the template. The monograph therefore records the language as a candidate framework, whose explanatory reach is Hypothesis 26.10, and the synthesis theorem below states only what the template implies once its three ingredients are given.

## 46.4 Formal Statements

**Theorem 46.1** (Conditional Geometric Lower Bound)**.** *Let $\varphi$ be a spectral point, $T$ a tensor with $\varphi(T) > 0$, and suppose that every circuit of size $L \geq 2$ computing the polynomial associated with $T$ yields an inequality $\log \varphi(T) \leq A + B + \eta \log L$ with constants $A, B \geq 0$ and $\eta \geq 0$ that do not depend on the circuit. If $\log \varphi(T) - A - B \geq \Delta > 0$, then $\eta > 0$ and every such circuit has size $L \geq 2^{\Delta/\eta}$. If $\eta = 0$ the hypotheses are inconsistent, so no circuit of that kind exists.*

*Proof.* From the inequality, $\eta \log L \geq \log \varphi(T) - A - B \geq \Delta > 0$. If $\eta = 0$ the left side is zero, a contradiction, so either no circuit exists or the hypotheses fail. If $\eta > 0$, then $\log L \geq \Delta/\eta$, that is $L \geq 2^{\Delta/\eta}$. □

**Corollary 46.2** (The Flat-Case Instance)**.** *In the setting of Proposition 15.5, suppose that for every $S \in \mathcal{B}_k$ the constants $A = k$, $B_S = \ell \sum_{(a,b)} d((a,b), h_S(a,b))$ and $\eta_S = d(o_S, o'_S) > 0$ satisfy $\ell \sum_{a<b} d((a,b),(b,a)) - A - B_S \geq \Delta > 0$. Then $L \geq 2^{\Delta/\max_S \eta_S}$, and since each $\eta_S \leq 1$ by Inherited Result 9.3, $L \geq 2^{\Delta}$.*

*Proof.* By Proposition 15.5 a circuit of size $L$ yields some $S$ with $\log \varphi(T_{k,\ell}) \leq k + B_S + \eta_S \log L$. Theorem 46.1 applied with that $S$ gives $L \geq 2^{\Delta/\eta_S} \geq 2^{\Delta/\max_S \eta_S}$. Since $\eta_S \leq 1$ the bound is at least $2^{\Delta}$. □

*Remark* 46.3. The mechanism is therefore an admissible decomposition, a spectral invariant and a geometric gap, together implying a resource lower bound. Corollary 46.2 is the template in the flat case, in which the gap is established by placing the ends at the centres of the facets of a cube and the inherited ingredients are those of Chapter 16. The substantive task in a new application is the independent establishment of the three ingredients, and the template contributes only the arithmetic that combines them.

# Chapter 47

# The Independence of Mathematical Knowledge

## 47.1 Independence of Result and History

A mathematical result can be accepted from an unknown or disputed source once its proof has been checked, and this is a consequence of the extensionality of the verification relation. A deterministic verifier applied to a statement and a certificate returns the same verdict whichever history produced the certificate. Chapter 28 used this to show that verification cannot serve as evidence about provenance, and the present chapter records the same fact as a statement of independence: the standing of a result as verified does not depend on how it was found, by whom, or in what sequence of attempts.

## 47.2 Portability Between Environments

Independence from discovery history is one half of the independence of knowledge. The other half is independence from a particular verification environment. A result that can be verified only in one system, on one machine, or through one tool is dependent on that environment in a way that a result with several independent verifications is not. Portability is the ability to carry an artifact from one environment to another while preserving what matters, namely the intended mathematical claims and the certificates that support them. It requires a translation, and the translation must be checked for fidelity as well as for acceptance, since Chapter 27 showed that a proof can be accepted by a kernel for a statement that does not express the intended claim.

## 47.3 The Conditions of Independence

Independence in both senses is a property of an artifact together with an infrastructure. The statement of the result must be fixed and unambiguous, the certificate must be checkable by

a mechanism that does not depend on the producer, the dependencies must be recorded, and the translation between environments must be faithful. These conditions can fail without any error in the mathematics, and a result that satisfies none of them is not thereby false. The monograph treats them as the engineering counterpart of the logical independence of proof from provenance, and its ledger and reproducibility records are the means by which they are checked.

## 47.4 Formal Statements

**Definition 47.1** (Artifact Portability)**.** Let $W$ be a mathematical artifact, $c$ an informal claim of $W$, and $\mathcal{E}_1, \mathcal{E}_2$ verification environments. A translation $\tau_{\mathcal{E}_1 \to \mathcal{E}_2}$ sending a pair $(\Theta_1, \pi_1)$ of a formal statement and a certificate in $\mathcal{E}_1$ to a pair $(\Theta_2, \pi_2)$ in $\mathcal{E}_2$ is portable for $c$ if it preserves acceptance, in that $\mathcal{K}_1(\Theta_1, \pi_1) = 1$ implies $\mathcal{K}_2(\Theta_2, \pi_2) = 1$, and it preserves fidelity, in that $\Theta_1$ faithful for $c$ implies $\Theta_2$ faithful for $c$ in the sense of Chapter 27.

**Proposition 47.2** (Independence from Discovery History)**.** *If $F(h_1) = F(h_2) = \pi$, then $V(T, F(h_1)) = V(T, F(h_2))$ for every statement $T$ and every deterministic verifier $V$.*

*Proof.* The arguments of $V$ are equal, so its values are equal. □

**Proposition 47.3** (Acceptance Alone Is Not Portability)**.** *There is a translation that preserves acceptance and is not portable for a claim $c$.*

*Proof.* Let $\mathcal{E}_2$ accept a proof $\pi_0$ of the formal statement $0 = 0$, and let $\tau$ send every pair to $(0 = 0, \pi_0)$. Acceptance is preserved. For a claim $c$ not expressed by $0 = 0$, such as the claim that every field has characteristic zero, $0 = 0$ is not faithful for $c$, so a faithful $\Theta_1$ is sent to an unfaithful $\Theta_2$ and fidelity is not preserved. This is Proposition 27.4 stated for translations. □

**Proposition 47.4** (Portable Translations Compose)**.** *If $\tau_{12}$ is portable for $c$ from $\mathcal{E}_1$ to $\mathcal{E}_2$ and $\tau_{23}$ is portable for $c$ from $\mathcal{E}_2$ to $\mathcal{E}_3$, then $\tau_{23} \circ \tau_{12}$ is portable for $c$ from $\mathcal{E}_1$ to $\mathcal{E}_3$.*

*Proof.* If $\mathcal{K}_1(\Theta_1, \pi_1) = 1$ then $\mathcal{K}_2$ accepts the image under $\tau_{12}$, and then $\mathcal{K}_3$ accepts its image under $\tau_{23}$. If $\Theta_1$ is faithful for $c$ then so is $\Theta_2$, and then $\Theta_3$. □

**Proof Obligation 47.5.** *Construct a portable translation of the formal development of the source from its present environment to an independent one, and check the fidelity of the translated statements by a human reader, in the sense of Obligation 27.11.*

# Chapter 48

# The Continuity of Intellectual Agency

## 48.1 A Maintained Research State

Chapters 29 to 31 treated continuation, authorship and control as relations among execution instances, artifacts and parties. The present chapter combines them into the description of a research programme in time. At each time the researcher has a state consisting of the current artifacts, the available knowledge, the dependency records and the unresolved questions. The state changes by updates, and an update is authorized when it is made by the researcher or under rights the researcher has conferred. The continuity of the research is the continuity of this sequence of states under authorized updates.

## 48.2 Instruments That Preserve Continuation

A computational instrument produces outputs, and each output must be incorporated into the state if it is to be used. An instrument supports continuity when its outputs can be incorporated without destroying the dependencies, the provenance, the correction rights or the ability to resume the work. The condition is not a property of the instrument in isolation. An instrument whose outputs are delivered with their provenance and can be inspected, rejected and revised under the researcher’s control supports continuity, and an instrument whose outputs replace parts of the state without a record, or whose use alters the sequence of questions according to a schedule that the researcher does not control, does not.

## 48.3 The Space of Authorized Continuations

The set of states reachable from a given state by sequences of authorized updates is the space of continuations available to the researcher. A computational instrument enlarges the space if it adds authorized updates that the researcher could not otherwise perform at

comparable expense. It diminishes the space if it adds updates that are not authorized, or if it removes authorized ones by consuming the resources that they require. The distinction is between amplification and substitution, and the normative conclusion of the chapter is that assistance ought to amplify. The conclusion is a proposal about the design and use of instruments, and the elementary propositions below fix the formal sense of enlargement.

## 48.4 Formal Statements

**Definition 48.1** (Maintained Research State)**.** A maintained research state is $S_t = (W_t, K_t, D_t, \mathcal{Q}_t)$, where $W_t$ is the set of current artifacts, $K_t$ the available knowledge, $D_t$ the dependency records and $\mathcal{Q}_t$ the unresolved questions.

**Definition 48.2** (Research Continuation)**.** Let $\mathrm{Auth}(a)$ be a set of updates, each a partial function from states and events to states, that are authorized by the researcher $a$. A continuation is a sequence $S_{t+1} = \mathcal{U}_a(S_t, e_t)$ with $\mathcal{U}_a \in \mathrm{Auth}(a)$ for every $t$. The continuation space from $S$ is $\mathrm{Reach}_a(S)$, the set of states reachable from $S$ by finite sequences of events and authorized updates.

**Proposition 48.3** (Enlargement of the Continuation Space)**.** *If* $\mathrm{Auth}(a) \subseteq \mathrm{Auth}'(a)$ *then* $\mathrm{Reach}_a(S) \subseteq \mathrm{Reach}'_a(S)$ *for every* $S$*. If* $\mathrm{Auth}(a)$ *consists only of the identity update, then* $\mathrm{Reach}_a(S) = \{S\}$*.*

*Proof.* A sequence of updates in $\mathrm{Auth}(a)$ is a sequence in $\mathrm{Auth}'(a)$, so every reachable state of the first space is reachable in the second. If only the identity is authorized, every sequence returns the same state. □

**Definition 48.4** (Continuation-Preserving Instrument)**.** An instrument $\mathcal{I}$ with output $o$ supports research continuity at $S = (W, K, D, \mathcal{Q})$ if there is an authorized update $\mathcal{U}_a$ incorporating $o$ into a state $S' = (W', K', D', \mathcal{Q}')$ such that the dependency records are not reduced, $D \subseteq D'$, the provenance of $o$ is recorded in $D'$, the control rights over the artifacts of $W'$ that the researcher held in $S$ are retained, and $\mathcal{U}_a$ is again defined at $S'$, so that the work can be resumed.

**Proposition 48.5** (Preservation Is Not Implied by Output Quality)**.** *There is an instrument all of whose outputs are correct and which does not support research continuity.*

*Proof.* Let an instrument return a correct proof $o$ of a statement together with a procedure that, on incorporation, replaces the dependency records by an empty set. Every output is correct, and every incorporation reduces $D$ to $\varnothing \not\supseteq D$ whenever $D \neq \varnothing$, so Definition 48.4 fails. Correctness of an output and preservation of the state into which it is incorporated are independent properties. □

**Principle 48.6** (Amplification and Not Substitution)**.** Computational assistance should enlarge the researcher's continuation space, in the sense of Proposition 48.3, and should not substitute an externally controlled sequence of objectives for the researcher's own. This is a

normative proposal. It is consistent with the calculator principle of Chapter 29 and with the principle that responsibility follows control of Chapter 31, and it is not derived from them.

# Chapter 49

# The Right Not to React

## 49.1 Deferral as an Operation

An event arrives at a time, and the party who receives it may process it at once or at a later time. Deferral is the operation that retains the event and changes the time at which it is processed. The value of processing at a given time is the benefit function of Chapter 32, and the loss from deferral by an interval is the delay sensitivity, the difference between the benefit now and the benefit after the interval. Both are defined without reference to who sent the event or why, and the analysis below uses only these quantities and the interruption cost of Chapter 33.

## 49.2 Deferral Without Loss

If the benefit of processing an event is unchanged over the interval, then deferral over that interval loses nothing, and the surplus of interrupting productive activity to process the event immediately is the negative of the interruption cost. When that cost is positive the surplus is negative. This is the content of the result that importance without urgency never justifies interruption, restated for the operation of deferral. Many events that are important, in the sense that their value is large, are not urgent, in the sense that their value does not decay over the relevant interval, and for them the rational course on the expenditure model is to decline to react at once.

## 49.3 Batching Deferred Events

When several deferrable events are deferred and processed together, the fixed components of the interruption cost, suspending the current activity and restoring it, are paid once and not once for each event. The saving is exactly the number of events less one times the fixed cost. The proposition below states this, and it shows that the benefit of declining to react is not only the avoidance of a single interruption but a reduction in the total cost of attending to the whole set of events.

## 49.4 A Conclusion Without Claims About Motives

The analysis characterizes the decision to defer in terms of benefit, delay sensitivity and interruption cost. It does not depend on any claim about why an event was produced, whether by an institution, a product or a person, and it makes no such claim. The conclusion that declining to react can be expenditure-minimizing follows from the structure of the costs alone. Whether any particular source of events is designed to convert importance into the appearance of urgency is an empirical question of the kind recorded in Obligation 35.9, and the formal result is the same whatever the answer.

## 49.5 Formal Statements

**Definition 49.1** (Deferral Operator)**.** For an event $e$, a time $t$ and $\Delta > 0$, the deferral operator is $D_\Delta(e, t) = (e, t + \Delta)$.

**Proposition 49.2** (Deferral Without Loss)**.** *If $B(e, t) = B(e, t + \Delta)$ then $U_\Delta(e, t) = 0$. If in addition the interruption cost satisfies $C_I(x_t, e) > 0$, then $S(x_t, e) = -C_I(x_t, e) < 0$, so immediate processing has negative surplus.*

*Proof.* By Definition 32.2, $U_\Delta(e, t) = B(e, t) - B(e, t + \Delta) = 0$. Then Proposition 33.3 gives $S(x_t, e) = -C_I(x_t, e)$, which is negative when $C_I > 0$. □

**Corollary 49.3** (Rational Nonreaction)**.** *When an event is deferrable without loss and immediate processing disrupts productive activity, so that $C_I(x_t, e) > 0$, declining to react immediately is the surplus-maximizing decision in the sense of Definition 33.2.*

*Proof.* The surplus of interrupting is negative by Proposition 49.2, and the surplus of not interrupting, which defers by $\Delta$ at no loss and at no interruption cost, is zero. □

**Proposition 49.4** (Batching Deferred Events)**.** *Let $n \geq 1$ deferrable events each have interruption cost $C_{\text{suspend}} + C_{\text{process},i} + C_{\text{restore}}$. Processing them one at a time, each by an interruption, costs $n(C_{\text{suspend}} + C_{\text{restore}}) + \sum_i C_{\text{process},i}$. Processing them together in a single interruption costs $C_{\text{suspend}} + C_{\text{restore}} + \sum_i C_{\text{process},i}$. The saving is $(n-1)(C_{\text{suspend}} + C_{\text{restore}}) \geq 0$, with strict inequality if $n \geq 2$ and $C_{\text{suspend}} + C_{\text{restore}} > 0$.*

*Proof.* Subtract the second expression from the first. □

# Summary of Original Contributions

The monograph has two complementary mathematical arcs. The first investigates how the organization of dependencies makes certain computations resistant to decomposition. The second investigates how the preservation of completed structures allows researchers to avoid repeating unnecessary computational and intellectual expenditure. The central unifying question is what must be preserved when a complex process is decomposed, abstracted, delegated or resumed. Five results organize the contributions of the later parts, and each is elementary at its foundations. Their originality, to the extent they have any, lies in their integration, interpretation and subsequent development, and not in the discovery of new inequalities or logical propositions.

The first is certificate-conditioned tractability and amortization. The average expenditure after $N$ uses is

$$\frac{C(N)}{N} = \frac{C_D + C_V + C_I + C_M(N)}{N} + \overline{C}_A,$$

where $\overline{C}_A$ is the mean application expenditure. This gives a formal account of the transformation of expensive discovery into reusable knowledge, with the break-even condition of Proposition 26.2 and the asymptotic statement of Theorem 45.1.

The second is provenance non-identifiability. If $F(h_1) = F(h_2)$ with $h_1 \neq h_2$, no function recovers the history from the artifact in both cases (Proposition 28.2), and the best rate of recovery is $|F(\mathcal{H})|/|\mathcal{H}|$ (Proposition 28.3). This separates mathematical certification from claims about the history of discovery or the identity of a producer.

The third is preservation under abstraction. A quantity descends through a surjection $q$ if and only if it is constant on the fibers of $q$ (Theorem 38.2). This connects the failure of arithmetic lower bounds to transfer through Boolean evaluation (Proposition 22.8 and Proposition 38.6) with a general account of what survives a many-to-one description.

The fourth is resource-bounded geometric obstruction. If a multiplicative monotone satisfies $\varphi(x) \leq 2^{\alpha R}\varphi(y)$ whenever $x$ is obtained from $y$ at cost $R$, then $R \geq \frac{1}{\alpha}\log_2(\varphi(x)/\varphi(y))$ (Corollary 40.6). This is a common framework for tensor restrictions, modular architectures and interface complexity, with the substantive geometric content remaining the hypothesis of Chapter 40.

The fifth is sustainable continuation under resource constraints. Under convex expendi-

ture the output over an interval of length $T$ satisfies

$$\int_0^T u(t)\,dt \le T^{1-1/p}\Big(\int_0^T u(t)^p\,dt\Big)^{1/p},$$

with equality for constant effort (Proposition 34.2). Together with the contextual non-identifiability of interruptions (Proposition 33.4) this supplies a formal foundation for the discussion of pacing, attention and the right not to react.

The mathematical core, the separation of the algebraic complexity classes, is an inherited result and has not been independently reconstructed in full. The crossing identity, spectral duality, the curvature theorem, the machine constructions of the contracting case, the field-transfer arguments and the border-complexity extension keep inherited status in the ledger until they have been reconstructed or independently verified.

# Appendix A

# Mathematical Preliminaries

## A.1 Conventions

The logarithm is to base two throughout, fields are arbitrary unless a characteristic is specified, and finite sets are written with vertical bars for their cardinality. A matrix over a field has a rank, the dimension of its column space, and the rank of the zero matrix is zero. A flattening of a tensor is the matrix obtained by grouping its indices into rows and columns in a stated order, and its rank does not depend on the order within each group, since reordering the indices within a group permutes the rows or the columns.

## A.2 Matrix Rank

Three facts about matrix rank are used repeatedly. The rank is subadditive under sums. It does not increase under multiplication by matrices of compatible size on either side. And it is multiplicative under the Kronecker product, which is the fact on which the cut-rank identity of Chapter 9 and the spectral point defined by a flattening in Chapter 8 depend. The first two are standard and are proved below in a line each. The third is stated in the literature and is proved here so that the cut-rank identity does not rest on an unproved citation.

## A.3 Elementary Inequalities

Four inequalities of analysis are used. The subadditivity of the power $x \mapsto x^a$ for $0 < a \leq 1$ is used in Chapter 8 to show that a power of a spectral point is a spectral point. The inequality of Cauchy and Schwarz for finite sums is used in the Euclidean estimates of Chapter 16. Jensen's inequality for a convex function is used to derive the form of Hölder's inequality that Chapter 34 requires, and the monotonicity of the logarithm is used throughout.

## A.4 Formal Statements

**Lemma A.1** (Rank of Sums and Products)**.** *For matrices of compatible sizes over a field,* $\operatorname{rank}(M+N) \le \operatorname{rank} M + \operatorname{rank} N$ *and* $\operatorname{rank}(PMQ) \le \operatorname{rank} M$*.*

*Proof.* The column space of $M+N$ is contained in the sum of the column spaces of $M$ and $N$, whose dimension is at most the sum of the dimensions. The column space of $PMQ$ is contained in the image under $P$ of the column space of $M$, whose dimension is at most $\operatorname{rank} M$. □

**Lemma A.2** (Rank Under the Kronecker Product)**.** *For matrices $M$ and $N$ over a field,* $\operatorname{rank}(M \otimes N) = \operatorname{rank} M \cdot \operatorname{rank} N$*.*

*Proof.* If either matrix is zero both sides are zero. Otherwise write $M = XY$ with $X$ of full column rank $r$ and $Y$ of full row rank $r$, which is a rank factorization, and $N = X'Y'$ with ranks $r'$ similarly. By the mixed-product property, $M \otimes N = (X \otimes X')(Y \otimes Y')$. A matrix of full column rank has a left inverse, so $X$ has a left inverse $L$ and $X'$ a left inverse $L'$, and $L \otimes L'$ is a left inverse of $X \otimes X'$, which therefore has full column rank $rr'$. Likewise $Y \otimes Y'$ has full row rank $rr'$. A product of a matrix of full column rank and a matrix of full row rank, with the inner dimension equal to both ranks, has that rank, so $\operatorname{rank}(M \otimes N) = rr'$. □

**Lemma A.3** (Subadditivity of Powers)**.** *For $0 < a \le 1$ and $x_1, \dots, x_n \ge 0$, $\left(\sum_j x_j\right)^a \le \sum_j x_j^a$.*

*Proof.* For two terms with $x + y > 0$, put $s = x/(x+y) \in [0,1]$. Since $t^a \ge t$ on $[0,1]$, $s^a + (1-s)^a \ge s + (1-s) = 1$, and multiplying by $(x+y)^a$ gives $x^a + y^a \ge (x+y)^a$. The general case follows by induction on $n$. □

**Lemma A.4** (Jensen Form of Hölder's Inequality)**.** *For $p > 1$, $T > 0$ and a nonnegative measurable function $u$ on $[0,T]$ with $\int_0^T u^p < \infty$, $\int_0^T u\,dt \le T^{1-1/p}\left(\int_0^T u^p\,dt\right)^{1/p}$, with equality if and only if $u$ is almost everywhere constant.*

*Proof.* The function $t \mapsto t^p$ is strictly convex on $[0,\infty)$. By Jensen's inequality for the uniform probability measure on $[0,T]$, $\left(\frac{1}{T}\int_0^T u\right)^p \le \frac{1}{T}\int_0^T u^p$, with equality exactly when $u$ is almost everywhere constant. Taking $p$th roots and multiplying by $T$ gives the claim. □

**Lemma A.5** (Cauchy–Schwarz for Finite Sums)**.** *For reals $c_1, \dots, c_k$, $\left(\sum_a c_a\right)^2 \le k \sum_a c_a^2$.*

*Proof.* This is the inequality of Cauchy and Schwarz for the vectors $(c_a)$ and $(1, \dots, 1)$. □

# Appendix B

# Arithmetic Circuit Definitions

## B.1 Circuits as Lists

The monograph uses a single circuit model, defined in Chapter 3 and in the statement file of the source: a circuit is a finite list of polynomials in which each entry is a variable, a constant of the field, or the sum or product of two earlier entries, and a polynomial is computed by the circuit if it occurs in the list. The size is the length of the list, so that variables and constants count as entries. This convention is the one the source uses, and the count of operations alone, which some treatments prefer, differs from it by at most a factor of three, as Proposition 27.9 shows. All lower bounds in this monograph are bounds on the length of the list.

## B.2 Elementary Properties of Size

Four elementary properties of the size are needed. A circuit can be padded to any larger length by appending variables, so the statement that every circuit for a polynomial has more than $s$ entries is monotone in $s$. The size of a sum or a product is at most the sum of the sizes plus one, which is the composition bound of Chapter 4. Replacing a variable by a constant does not increase the size. And the degree of the polynomial computed by an entry grows at most exponentially with its position, so that the size of any circuit for a polynomial of degree $\delta$ is at least logarithmic in $\delta$.

## B.3 Families and Classes

A family of polynomials has one polynomial for each size parameter. The class VP consists of families with polynomially many variables, polynomially bounded degree and polynomially bounded circuit size, and VNP consists of families that are exponential sums over Boolean assignments of the values of a family in VP, as defined in Chapter 4. The permanent and the Hamiltonian cycle polynomial are the families on which the lower bounds are proved,

and both belong to VNP, the permanent by its definition as a sum over permutations that can be written as an exponential sum, and the Hamiltonian cycle polynomial likewise.

## B.4 Boolean and Approximating Variants

Two variants of the model are used in Part VIII and in Chapter 24. The Boolean quotient cost is the least size of a circuit among all polynomials with a given Boolean evaluation, and it is at most the size of any one of them, so that a lower bound on the polynomial size does not bound it. The border cost allows the circuit to be a circuit over the field of formal Laurent series computing a polynomial that approximates the target in the sense of Definition 24.1. These variants are defined where they are used, and their relation to the model above is the subject of Chapters 22 and 24.

## B.5 Formal Statements

**Definition B.1** (Circuit and Size)**.** A circuit over a field $\mathbb{F}$ in the variables $x_1, \dots, x_n$ is a list $(g_1, \dots, g_L)$ of polynomials such that each $g_j$ is a variable, an element of $\mathbb{F}$, or $g_i + g_{i'}$ or $g_i g_{i'}$ for some $i, i' < j$. It computes $f$ if $f = g_j$ for some $j$. Its size is $L$, and $L_{\mathbb{F}}(f)$ is the least size of a circuit computing $f$.

**Proposition B.2** (Degree Bound)**.** *In a circuit $(g_1, \dots, g_L)$ one has $\deg g_j \le 2^{j-1}$ for every $j$. Hence if the circuit computes a polynomial $f$ of degree $\delta \ge 1$, then $L \ge 1 + \log \delta$.*

*Proof.* For $j = 1$ the entry is a variable or a constant, of degree at most $1 = 2^0$. If $g_j$ is a variable or a constant its degree is at most $1 \le 2^{j-1}$. If $g_j = g_i + g_{i'}$ its degree is at most $\max\{\deg g_i, \deg g_{i'}\} \le 2^{j-2}$ by induction. If $g_j = g_i g_{i'}$ its degree is at most $2^{i-1} + 2^{i'-1} \le 2 \cdot 2^{j-2} = 2^{j-1}$. If $f = g_j$ then $\delta \le 2^{j-1} \le 2^{L-1}$, so $L \ge 1 + \log \delta$. □

**Proposition B.3** (Padding and Monotonicity)**.** *If $f$ is computed by a circuit of size $s$, it is computed by a circuit of every size $s' \ge s$. Hence $\{s : L_{\mathbb{F}}(f) \le s\}$ is an up-set of the naturals.*

*Proof.* Append $s' - s$ copies of a variable. The result is a circuit, and $f$ still occurs in it. □

**Proposition B.4** (Substitution of Constants)**.** *For a variable $x_i$ and $c \in \mathbb{F}$, $L_{\mathbb{F}}(f|_{x_i=c}) \le L_{\mathbb{F}}(f)$.*

*Proof.* In a circuit computing $f$, replace every entry equal to the variable $x_i$ by the constant $c$, and replace each later entry by the same operation on the replaced operands. By induction on the position, each new entry is the old entry with $x_i$ set to $c$, so $f|_{x_i=c}$ occurs in the new list of the same length. □

**Proposition B.5** (Renaming of Variables)**.** *If $\rho$ is a bijection of the variables and $f^\rho$ is the polynomial obtained by renaming, then $L_{\mathbb{F}}(f^\rho) = L_{\mathbb{F}}(f)$. In particular the permanent of the*

*transpose of a generic matrix and the permanent of the matrix itself have the same circuit size.*

*Proof.* Renaming every variable entry of a circuit by $\rho$ gives a circuit of the same length for $f^\rho$, and $(f^\rho)^{\rho^{-1}} = f$ gives the reverse inequality. The permanent of the transpose is the permanent of the generic matrix with $X_{i,j}$ renamed to $X_{j,i}$, a bijection of the variables. □

# Appendix C

# Spectral Functionals and Tensor Restrictions

## C.1 Reconstruction of the Spectral Exponents

The pseudometric attached to a spectral point in Chapter 9 is the lemma of the source that converts the abstract axioms into geometry, and it is the first inherited result of the geometric argument. The manuscript's argument is short, and it uses only the axioms of Chapter 8 and the wire composition of Chapter 9. This appendix writes it out completely. A function of the integers that is multiplicative and nondecreasing is a power, by an argument of classical type that is reproduced below. The wire exponent is such a function, so it is a power, with an exponent between zero and one by the cover axiom. The triangle inequality is a restriction of a product of two wires to a third.

## C.2 What the Reconstruction Uses

The reconstruction uses the definition of a spectral point as a normalized multiplicative function subadditive on covers, the preorder of local restriction, the composition of wires up to local change of basis, and nothing else. It does not use the existence or duality theorems, which concern the supply of spectral points, and it applies to every spectral point, whatever its provenance. This is why the exponent lemma can be proved here while the existence of spectral points is inherited, and the ledger records the two differently.

## C.3 Formal Statements

**Lemma C.1** (Multiplicative Monotone Functions on the Integers)**.** *Let $g : \{1, 2, 3, \dots\} \to [1, \infty)$ satisfy $g(pq) = g(p)g(q)$ for all $p, q$ and $g(p) \le g(q)$ for $p \le q$. Then $g(q) = q^c$ for all $q$, where $c = \log g(2) \ge 0$.*

*Proof.* From $g(1) = g(1)^2$ and $g(1) \ge 1$ we get $g(1) = 1$. For $q \ge 2$ and $n \ge 1$ choose

$m \geq 0$ with $2^m \leq q^n < 2^{m+1}$. By monotonicity and multiplicativity, $g(2)^m = g(2^m) \leq g(q^n) = g(q)^n \leq g(2^{m+1}) = g(2)^{m+1}$. Taking logarithms, $mc \leq n \log g(q) \leq (m+1)c$. Also $m \leq n \log q < m+1$, so $mc \leq nc \log q < (m+1)c$. Hence $|n \log g(q) - nc \log q| \leq c$, that is $|\log g(q) - c \log q| \leq c/n$. Letting $n \to \infty$ gives $\log g(q) = c \log q$, that is $g(q) = q^c$. □

**Proposition C.2** (Reconstruction of the Wire Exponents)**.** *Let $\varphi$ be a spectral point on $V$ and $u \neq v$ parties. Then there is $d_\varphi(u,v) \in [0,1]$ with $\varphi(W_q(u,v)) = q^{d_\varphi(u,v)}$ for all $q \geq 1$. Setting $d_\varphi(u,u) = 0$, the function $d_\varphi$ is symmetric and satisfies $d_\varphi(u,w) \leq d_\varphi(u,v) + d_\varphi(v,w)$ for all $u, v, w$.*

*Proof.* Put $g(q) = \varphi(W_q(u,v))$. Multiplicativity: by Proposition C.3 below, $W_p W_q \simeq W_{pq}$, and restriction-equivalent tensors have equal values by monotonicity, so $g(pq) = \varphi(W_p W_q) = g(p)g(q)$. Monotonicity: for $q \leq q'$ the maps $P : \mathbb{F}^{q'} \to \mathbb{F}^q$ that keep the first $q$ coordinates, applied at $u$ and at $v$, send $W_{q'}(u,v)$ to $W_q(u,v)$, so $W_q \preceq W_{q'}$ and $g(q) \leq g(q')$. Lower bound: $W_q \neq 0$, so $g(q) \geq 1$. By Lemma C.1, $g(q) = q^d$ with $d \geq 0$. Upper bound: $W_q(u,v) = \sum_{i=1}^q e_i^{(u)} \otimes e_i^{(v)}$ is a sum of $q$ tensors each a restriction of $\mathbf{1}$, since a tensor with one nonzero index at each of $u, v$ and one-element index sets elsewhere is the image of $\mathbf{1}$ under the maps $1 \mapsto e_i$ at $u$ and $v$. The cover axiom gives $g(q) \leq q\varphi(\mathbf{1}) = q$, so for $q \geq 2$, $q^d \leq q$, hence $d \leq 1$. Symmetry holds because $W_q(u,v) = W_q(v,u)$. For the triangle inequality with $u, v, w$ distinct, the product $W_q(u,v)W_q(v,w) = \sum_{i,j} e_i^{(u)} \otimes (e_i \otimes e_j)^{(v)} \otimes e_j^{(w)}$ has index space $\mathbb{F}^q \otimes \mathbb{F}^q$ at $v$. The linear map at $v$ sending $e_i \otimes e_j$ to $\delta_{ij}$ in $\mathbb{F}^1$ carries it to $\sum_i e_i^{(u)} \otimes e_i^{(w)} = W_q(u,w)$. Hence $W_q(u,w) \preceq W_q(u,v)W_q(v,w)$, and $q^{d(u,w)} \leq q^{d(u,v)+d(v,w)}$ for $q \geq 2$ gives the inequality. If two of the three points coincide the inequality is immediate from symmetry and $d(u,u) = 0$. □

**Proposition C.3** (Wire Composition)**.** *For parties $u \neq v$, $W_p(u,v)W_q(u,v) \simeq W_{pq}(u,v)$.*

*Proof.* This is the multiplicative wire composition of Chapter 9, whose proof identifies $\mathbb{F}^p \otimes \mathbb{F}^q$ with $\mathbb{F}^{pq}$ at each of the two parties by a basis-matching isomorphism, and isomorphisms are restrictions in both directions. □

**Corollary C.4** (Status of the Exponent Lemma)**.** *Inherited Result 9.3 follows from the axioms of a spectral point and Proposition C.3, as shown in Proposition C.2, and does not depend on the existence or duality theorems.*

# Appendix D

# Metric Geometry

## D.1 Pseudometrics and Their Quotients

The functions $d_\varphi$ are pseudometrics, and distinct parties can be at distance zero. The geometric constructions of Chapter 12 work with metric spaces, so a quotient is needed. Identifying points at distance zero produces a metric space, and the distance is well defined on the quotient because the triangle inequality shows that it does not depend on the representatives. The operation is elementary, and it is one of several that Chapter 12 separates when it cautions that the enlarged space must again be a price.

## D.2 Midpoints and the Curvature Inequality

The curvature inequality in the form of Bruhat and Tits compares the squared distance from an arbitrary point to the midpoint of two points with the average of the squared distances to the two points. In a Euclidean space the comparison is an identity, by the parallelogram law. In a space of nonpositive curvature it is an inequality, and the margin of Chapter 43 measures the difference. Two forms of the inequality occur in the manuscript and in this monograph, one with a midpoint of two points at a given distance and one with a pair of points at distance $2D$ and a point at distance at most $D$ from each. They are the same statement, as the first lemma below shows.

## D.3 Consequences Used Later

Two consequences of the inequality are used in the later chapters. The distance from a point to the midpoint of two points is at most the larger of the distances to the two, a form of convexity that is used in the induction of the facet-centres proposition. And in a metric space midpoints are unique when the inequality holds, as Chapter 12 proved. The substantive geometric results, the realization of prices as pull-backs of CAT(0) metrics and the supporting results of the source's Appendix O, remain inherited, and this appendix does not reconstruct them.

## D.4 Formal Statements

**Lemma D.1** (Quotient of a Pseudometric)**.** *Let $d$ be a pseudometric on a set $X$ and $x \sim y$ if $d(x,y) = 0$. Then $\sim$ is an equivalence relation and $\bar{d}([x],[y]) = d(x,y)$ is a well defined metric on the quotient $X/\!\sim$.*

*Proof.* Reflexivity and symmetry follow from $d(x,x) = 0$ and symmetry. If $d(x,y) = d(y,z) = 0$ then $d(x,z) \le 0$, so transitivity holds. If $x \sim x'$ and $y \sim y'$ then $d(x,y) \le d(x,x') + d(x',y') + d(y',y) = d(x',y')$ and symmetrically, so $\bar{d}$ is well defined. It satisfies the triangle inequality because $d$ does, and $\bar{d}([x],[y]) = 0$ implies $x \sim y$. □

**Lemma D.2** (Equivalence of the Two Forms of the Inequality)**.** *For points $x, v, m, z$ of a metric space with $d(x,m) = d(v,m) = D$ and $d(x,v) = 2D$, the inequality $d(z,m)^2 \le \frac{1}{2}d(z,x)^2 + \frac{1}{2}d(z,v)^2 - D^2$ is the statement $M(x,v,m,z) \ge 0$ of Definition 43.4.*

*Proof.* Here $m$ is a midpoint of $x$ and $v$ and $\frac{1}{4}d(x,v)^2 = \frac{1}{4}\cdot 4D^2 = D^2$. Substituting into the definition of $M$ gives the equivalence. □

**Lemma D.3** (Convexity of Distance to a Point)**.** *In a metric space satisfying $M \ge 0$ for all midpoints, if $y$ is a midpoint of $x$ and $x'$ then $d(z,y) \le \max\{d(z,x), d(z,x')\}$ for every $z$.*

*Proof.* By $M \ge 0$, $d(z,y)^2 \le \frac{1}{2}d(z,x)^2 + \frac{1}{2}d(z,x')^2 - \frac{1}{4}d(x,x')^2 \le \max\{d(z,x), d(z,x')\}^2$. □

**Lemma D.4** (Euclidean Equality)**.** *In a Euclidean space, $\|z-x\|^2 + \|z-x'\|^2 = 2\|z-y\|^2 + \frac{1}{2}\|x-x'\|^2$ for $y = \frac{1}{2}(x+x')$, so $M = 0$ for every $z$.*

*Proof.* Write $a = x - y = -(x'-y)$ and $b = z - y$. Then $\|z-x\|^2 = \|b-a\|^2$ and $\|z-x'\|^2 = \|b+a\|^2$, whose sum is $2\|b\|^2 + 2\|a\|^2$, and $\|a\|^2 = \frac{1}{4}\|x-x'\|^2$. □

**Proof Obligation D.5.** *Reconstruct the pull-back of a price to a complete CAT(0) space and the supporting results of Appendix O of the source, separating the quotient of Lemma D.1, the adjunction of points, the construction of midpoints and the metric completion, and showing at each stage that the metric remains a price.*

# Appendix E

# Reconstruction of the Flat-Case Proof

The derivations of the flat case are given in Chapters 14 to 16, and this appendix records their status. The following have been proved in the text: the structure of the network polynomial and its coefficient tensor (Proposition 14.2), the cut rank of the complete network (Proposition 14.3), the contrast with the path network (Proposition 14.6), the balanced decomposition of an arbitrary circuit (Proposition 15.2 and Corollary 15.3), the hub realization (Proposition 10.3), the simultaneous spectral inequality (Proposition 15.5), the Euclidean estimates (Proposition 16.2), the first lower bound under the flat condition (Proposition 16.4) and the conditional exclusion of polynomial size circuits (Theorem 16.5). The explicit bound (Proposition 16.8) and the final rate (Theorem 16.9) are proved from three inherited results: the sums-with-few-terms lemma, the facet-centres proposition, and the reduction of the network polynomial to the permanent. The transfer to the Hamiltonian cycle polynomial and the reduction to the field of the dichotomy remain inherited. The arithmetic of the constants was recomputed independently of the source for the permanent, and no machine verification of this recomputation has been carried out.

## E.1 The Chain of Inequalities

The flat-case argument has the structure of Theorem 46.1. A circuit for the network polynomial decomposes the coefficient tensor into at most $2^k$ pieces, each a restriction of a hub network (Proposition 15.2 and Corollary 15.3). The simultaneous spectral inequality (Proposition 15.5) follows from the cover axiom. Under the flat condition the exponent pseudometric is within a small error of the Euclidean metric on a placement of the ends at the centres of the facets of a cube, and the Euclidean estimates (Proposition 16.2) bound the distances from the ends to the hubs. The inherited lemma on sums with few terms replaces the factor $2^k$ by a smaller factor controlled by the exponent of the unit tensor on the ends, which the facet-centres proposition bounds by about $\sqrt{k}$. The resulting inequality gives $\log L \geq \ell k^{3/2}/14$ for $\ell \geq 100\sqrt{k}$ (Proposition 16.8), and the choice of $k = r^2$ and $\ell = 100r$

in terms of the size $m$ of the permanent gives the rate $2^{c(\log m)^{4/3}}$ (Theorem 16.9). The three inherited ingredients are the realization of the network polynomial as a permanent, the lemma on sums with few terms, and the facet-centres proposition together with the midpoint convexity that it uses.

## E.2 An Explicit Threshold

Theorem 16.9 asserts the existence of a threshold. The proof contains the inequalities from which an explicit threshold can be read, and the reading is a finite calculation. It is carried out below, relative to the inherited ingredients, and it shows that the threshold is $m_0 = 2^{1600}$. The calculation also shows how much weight the constant carries: at the threshold the bound is the trivial bound $L > 2$, and the theorem becomes informative only far beyond it. To exclude circuits of polynomial size $m^C$ the bound $2^{(\log m/1600)^{4/3}}$ must exceed $m^C$, which happens when $\log m > C^3 \cdot 1600^4$, that is, $\log m > 6.5536 \times 10^{12}\, C^3$. The result is a statement about asymptotics, and the practical accessibility of the numerical content is a separate question in the sense of Chapter 25.

## E.3 What Remains Inherited

The three inherited ingredients of the flat case keep their status in the ledger. Their reconstruction is the content of Obligation D.5 for the geometric part, of the reconstruction of the clause-sum realizations as permanents and Hamiltonian cycle polynomials, and of the sums-with-few-terms lemma. The reduction of the Hamiltonian cycle polynomial to the field of the dichotomy is also inherited. No machine verification has been carried out for any step of the chain, and the arithmetic of the explicit threshold below is a hand calculation that can be checked in a few lines.

**Proposition E.1** (Explicit Flat-Case Threshold)**.** *Assume the hypotheses of Theorem 16.9. Then the conclusion of the theorem holds with $m_0 = 2^{1600}$: for every $m \geq 2^{1600}$ every circuit for $\mathrm{PERM}_m$ has more than $2^{(\log m/1600)^{4/3}}$ gates.*

*Proof.* Let $m \geq 2^{1600}$, so that $\lfloor \log m \rfloor \geq 1600$, and let $r$ be the largest integer with $200r^3 \leq \lfloor \log m \rfloor$. Since $200 \cdot 2^3 = 1600$, $r \geq 2$. Put $k = r^2 \geq 4$ and $\ell = 100r = 100\sqrt{k}$. First, $3k^3\ell\, 2^{k\ell} = 300r^7\, 2^{100r^3}$. Since $r \leq 2^r$ we have $r^7 \leq 2^{7r}$, and $300 < 2^9$, so $300r^7 \leq 2^{7r+9} \leq 2^{100r^3}$ because $7r + 9 \leq 100r^3$ for $r \geq 1$. Hence $3k^3\ell\, 2^{k\ell} \leq 2^{200r^3} \leq 2^{\lfloor \log m \rfloor} \leq m$, which is the size condition of Inherited Result 14.7. Second, maximality of $r$ gives $\lfloor \log m \rfloor < 200(r+1)^3$, so $\log m < \lfloor \log m \rfloor + 1 \leq 200(r+1)^3 \leq 200 \cdot 8r^3 = 1600r^3$, using $r + 1 \leq 2r$. Third, Proposition 16.8 with $k \geq 2$ and $\ell = 100\sqrt{k}$ gives, for a circuit of $L$ gates, $\log(5L) \geq \ell k^{3/2}/16 = \frac{25}{4} r^4$. Since $r \geq 2$, $\frac{21}{4} r^4 \geq 84 > \log 5$, so $\log L \geq \frac{25}{4} r^4 - \log 5 > r^4$. Finally $r^3 > \log m/1600$ gives $r^4 > (\log m/1600)^{4/3}$, so $L > 2^{r^4} > 2^{(\log m/1600)^{4/3}}$. □

*Remark* E.2. Proposition E.1 resolves the flat-case part of Obligation 25.11. The contracting-case part, an explicit threshold for the cutting construction, remains open, and the source

does not make its constants explicit.

# Appendix F

# Reconstruction of the Contracting-Case Proof

## F.1 The Chain of Results

The contracting-case argument is longer than the flat-case argument and rests on a larger body of inherited results. The contracting hypothesis, a strict contraction of rank displacement on matrix spaces over the field with two elements, is converted by the Poincaré inequality on unitriangular groups (Chapter 17) into an inequality on the lengths of edges of Cayley graphs, and then, by a Cauchy–Schwarz step, into an inequality on spectral expenditure (Chapter 18). Existence of laws, the tower of groups and the tree decomposition (Chapter 18) convert an arbitrary circuit into a decomposition along a tree. The universal tape automaton and its runs as networks (Chapter 19) allow a circuit of small size to be embedded in the network structure, and computational cutting at all scales supplies the accelerators of Chapter 20. The diagonal argument of Chapter 21 shows that accelerators cannot exist at every scale and gives the lower bound $2^{c(\log m)^2/\log\log m}$, which dominates the flat-case rate.

## F.2 Status of Each Step

The table records the status of each step as in the ledger. Of the steps, the elementary ones, the reduction of the length-function and edge forms to the Poincaré inequality under the contracting condition, the bound on spectral expenditure, the finite repetition of squarings and the dominance of the contracting rate, are proved in the text. The inequalities from which the rest are derived, the Poincaré inequality itself, existence of laws, the tree decomposition, the machine constructions and the nonexistence of accelerators, are inherited, and several of these are verified only in the Lean development of the source. The monograph does not reconstruct them, and the lower bound of Theorem 21.7 is conditional on the two inherited results that it names.

| Step | Status | Location |
|---|---|---|
| Poincaré inequality on unitriangular groups | I | Chapter 17 |
| Length functions from prices | P(c) | Chapter 17 |
| Edge form of the Poincaré inequality | P(c) | Chapter 17 |
| Poincaré to spectral inequality | P(c) | Chapter 18 |
| Spectral expenditure bound | P | Chapter 18 |
| Existence of laws | I | Chapter 18 |
| Tower of groups and tree decomposition | I | Chapter 18 |
| Universal tape automaton, runs as networks | I | Chapter 19 |
| One cut and cuts at all scales | I | Chapter 19 |
| Accelerators from small circuits | I | Chapter 20 |
| Postponing constants and transfer to the next scale | I | Chapter 20 |
| Finite repetition of squarings | P | Chapter 21 |
| Nonexistence of accelerators | I | Chapter 21 |
| Contracting-case lower bound | P(c) | Chapter 21 |
| Dominance of the contracting rate | P | Chapter 21 |

## F.3 Open Obligations

Three obligations are attached to this branch. The Poincaré inequality and the supporting geometry must be reconstructed, using the extension and completion constructions that Chapter 12 describes. The machine constructions must be reproduced or audited in the formal development, since the printed proofs account for the programs at the level of exponents and state that the programs perform as claimed only in Lean. And an explicit threshold must be extracted from the constants of the cutting construction, which the source does not state. None of the three is attempted here.

# Appendix G

# Formal Verification Audit

## G.1 The Audit Register

This appendix collects the audit status of every verification claim that the monograph uses, in the form of Definition 27.7. Each claim is one of the source's reports, a comparison made in this monograph between the printed statement file and the text, or a claim about the formalization that has no support. The status is one of reported, reproduced or not checked, with the evidence stated. The entry reproduced means reproduced in this monograph, and none of the source's reports about the build or the checkers has that status.

| Claim | Status | Evidence | Remaining check |
|---|---|---|---|
| Toolchain is Lean 4.33.1 and Mathlib at the stated commit | Reported | Source Section 3 and Appendix Q | Rebuild at the cited commit |
| No `sorry` outside `Challenge.lean` | Reported | Source Appendix Q | Search of the built sources |
| Axioms of the theorems are `propext`, `Classical.choice`, `Quot.sound` | Reported | Source Section 3 | Print axioms of each theorem |
| No use of `native_decide` | Reported | Source Appendix Q | Search of the built sources |
| Proofs accepted by the Lean kernel | Reported | Source Section 3 | Rebuild |
| Proofs replayed by a second checker, `nanoda`, through a comparison tool | Reported | Source Section 3 | Rerun the comparison |

| Claim | Status | Evidence | Remaining check |
|---|---|---|---|
| Fidelity of the statement file to the intended claims | Not checked by a human reader | Source reports judgment by AI readers only | Human audit (Obligation 27.11) |
| Circuit model of the statement file equals that of Chapter 3 | Compared with printed text | Chapter 27 | Human audit of the file itself |
| Definition of approximation equals the Laurent definition | Compared with printed text | Proposition 27.10 | Human audit of the file itself |
| Permanent in the file is the permanent of the transpose | Compared with printed text | Chapter 27 and Proposition B.5 | None beyond the audit above |
| Gate-count translation by a factor of three | Not formalized in Lean | Source states it as not formalized; Proposition 27.9 proves it on paper | Formalize or audit |
| Remark N.6 on approximate determinantal complexity | No Lean equivalent | Source states this | None possible in Lean as it stands |
| Machine constructions behind Proposition I.1, parts of Theorem H.2, Theorem J.3 and Proposition N.4 | Verified in Lean only, per the source | Source | Reproduce or audit the formal declarations |
| Claim that the proofs were found and written by AI | Not checked | Source states it; Chapter 28 shows it is not entailed by the proofs | Evidence beyond the proof objects |

## G.2 The Four Principal Propositions

The source's statement file states four propositions: the main theorem for exact circuits, the main theorem for approximating circuits, the separation of VP from VNP over every field, and the statement that VNP is not contained in the closure of VP. On the evidence recorded above, the audit status vector of each of the four has the same components. The syntax component and the kernel component are reported. The axiom component is reported, with the footprint stated. The semantic component is not checked by a human reader. The replay component is reported. No component has been reproduced here. The four

propositions differ in which of the statements of the register they depend on, since the approximating-circuit statements also depend on the definition of approximation and the borderline statements on the machine constructions of Chapter 24.

## G.3 What a Human Reader Would Check

The audit that would raise the semantic component is limited and concrete, and it concentrates on the statement file and on the library definitions it uses, not on the proofs. The reader would check the definition of a circuit and its size, the definitions of the permanent and of the Hamiltonian cycle polynomial, the real exponent in the bound, the quantifier order, the definitions of VP, VNP and the closure of VP, and the library definitions of multivariate polynomials, Laurent series and cycles on a set. Chapter 27 records the comparisons already made against the printed text. The reader would then rebuild the development at the cited commit, confirm the absence of `sorry` outside the challenge file, print the axioms of each theorem, and rerun the comparison with the second checker. Each of these steps is recorded as an obligation and none has been performed.

# Appendix H

# Dependency Ledger

## H.1 Correspondence Between Source Results and Chapters

The table lists the results of the source manuscript that carry the mathematical core, the chapter of this monograph in which each is used, and its function. The status of each in this monograph is recorded in the ledger below. The source manuscript gives a corresponding mapping from its statements to Lean theorem names in Appendix Q, Table 3, which is reported there and not reproduced here.

| Source result | Chapter | Function |
|---|---|---|
| A.2 | 9 | Wire exponents and pseudometrics |
| A.3 | 10 | Rank-to-distance conversion |
| A.5 to A.6 | 8, 18 | Spectral existence and duality |
| B.2 to B.3 | 11 | Crossing identities |
| C.1 to C.4 | 12 | Nonpositive curvature |
| D.2 | 13 | Flat or contracting alternative |
| E.1 to E.3 | 14 to 16 | Completeness reductions |
| E.4 | 15 | Balanced decomposition |
| E.7 to E.11 | 16 | Quantitative flat-case bound |
| F.6 | 17 | Poincaré inequality |
| G.2 | 18 | Rerouting laws |
| H.2 | 18 | Tree decomposition |
| I.1 to I.2 | 19 | Universal computation networks |
| J.3 to J.4 | 19 | Computational cutting |
| K.2 to K.5 | 20 | Accelerator construction |
| K.6 to K.7 | 21 | Diagonal contradiction |
| L.1 | 6, 21 | Assembly of the separation |
| M.1 to M.4 | 6, 23 | Field and Hamiltonian transfer |
| N.2 to N.5 | 24 | Border complexity |
| O | 12, 13 | Supporting CAT(0) geometry |

## H.2 The Ledger

The ledger distinguishes statements proved in the text from inherited results, computational observations, and external references. The status column uses the categories of the Conventions chapter: P for a proposition proved here, P(c) for a proposition proved here conditionally on the inherited items named, I for an inherited result, C for a computational observation, H for a hypothesis or an open proof obligation, and Pr for a normative principle or proposal. References to lettered results are to the source manuscript.

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Certificate-conditioned tractability | P | Ch. 1 | Church's theorem for the undecidable projection. |
| Existence without availability | P | Ch. 1 | Definitions of knowledge environment and access cost only. |
| Amortized cost of shared knowledge | P | Ch. 2 | Elementary; rests on the stated accounting model. |
| Kernel of the Boolean evaluation map | P | Ch. 3 | Self-contained. |
| Quotient cost and lifting | P | Ch. 3 | Self-contained. |
| Completeness principle | P | Ch. 4 | Assumes completeness of the family, which is cited. |
| Valiant completeness theorems | I | Ch. 4 | Valiant 1979 and the exposition of Bürgisser [10]; cited, not reproved. |
| Determinant in VP | I | Ch. 5 | Berkowitz 1984; cited. |
| Principal separation (HC and PERM bounds) | I | Ch. 6 | Theorem M.3; not reconstructed in this version. |
| Existence and duality of spectral points | I | Ch. 8 | Propositions A.5, A.6; Zorn's lemma argument. |
| Spectral exponents and pseudometric | I, P | Ch. 9 | Lemma A.2; reconstructed in Appendix C from the spectral point axioms and wire composition, with existence of the spectral points remaining inherited. |
| Cut rank of graph tensors | P | Ch. 9 | Multiplicativity of matrix rank. |
| Hub realization | P | Ch. 10 | Two-hub version only; the version with given hubs is not reproved. |
| Crossing identity | I, C | Ch. 11 | Proposition B.2; exact finite checks on small cases, with the rank-bound check non-discriminating. |
| Crossing inequality | I | Ch. 11 | Proposition B.3. |
| Curvature inequality for prices | I | Ch. 12 | Theorem C.1. |
| CAT(0) realization of prices | I | Ch. 12 | Proposition C.3 and Lemma C.4. |
| Flat or contracting alternative | I | Ch. 13 | Theorem D.2; exhaustive, not shown exclusive. |
| Structure of the network polynomial | P | Ch. 14 | Self-contained. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Network polynomial as a permanent | I | Ch. 14 | Lemma E.3 with Proposition E.1, Corollary E.2, Proposition M.2. |
| Balanced products | P | Ch. 15 | Reconstruction of Lemma E.4; self-contained. |
| Simultaneous spectral inequality | P | Ch. 15 | Hub realization, cover axiom, spectral exponents (I). |
| Euclidean estimates | P | Ch. 16 | Elementary geometry; constants recomputed. |
| First lower bound for the network polynomial | P(c) | Ch. 16 | Flat condition, spectral exponents, pull-back of spectral points. |
| Explicit lower bound and final rate | P(c) | Ch. 16 | Lemmas E.7, E.8, Propositions E.9, A.7, Theorem C.1, Lemma E.3. |
| Transfer to HC and to the field of the dichotomy | I | Ch. 16 | Proposition M.2, Lemma M.1. |
| Poincaré inequality on unitriangular groups | I | Ch. 17 | Theorem F.6; relies on CAT(0) realization of prices and the contracting condition. |
| Length functions from prices | P(c) | Ch. 17 | Contracting condition; pull-back of prices. |
| Edge form of the Poincaré inequality | P(c) | Ch. 17 | Derived from Theorem F.6 by counting. |
| Poincaré to spectral inequality | P(c) | Ch. 18 | Cauchy–Schwarz; rounding of loads is left to the source. |
| Spectral expenditure bound | P | Ch. 18 | Elementary direction of duality. |
| Existence of laws | I | Ch. 18 | Theorem G.2; uses duality (Theorem A.6). |
| Tower of groups and tree decomposition | I | Ch. 18 | Proposition H.1, Theorem H.2; exactness of the straight-line program is in Lean only. |
| Universal tape automaton, runs as networks | I | Ch. 19 | Proposition I.1, Lemma I.2; machine constructions verified in Lean only. |
| One cut and cuts at all scales | I | Ch. 19 | Theorems J.3, J.4; exponents inherited, not rederived. |
| Evaluation identity and degree bound for $\Phi_{n,t}$ | P | Ch. 20 | Self-contained. |
| Accelerators from small circuits | I | Ch. 20 | Lemma K.3, Lemma K.2. |
| Postponing constants and transfer to the next scale | I | Ch. 20 | Lemma K.4, Proposition K.5; exponent accumulation recomputed. |
| Finite repetition of squarings | P | Ch. 21 | Pigeonhole. |
| Nonexistence of accelerators | I | Ch. 21 | Theorem K.6; the closing contradiction $V = 1 + V$ is elementary. |
| Contracting-case lower bound | P(c) | Ch. 21 | Lemma K.3, Theorem K.6; no explicit $m_0$. |
| Dominance of the contracting rate | P | Ch. 21 | Elementary asymptotics. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Boolean equivalence, failure of descent, image of the ideal | P | Ch. 22 | Proposition 3.5; elementary. |
| Nontransfer of lower bounds to the quotient cost | P | Ch. 22 | Self-contained; lift cost left open (Obligation 22.11). |
| Tractable projection closure and equal support | P | Ch. 23 | Elementary. |
| Exact determinantal complexity of the permanent | P(c) | Ch. 23 | Principal bound (I); Berkowitz (I). |
| No determinantal analogue of the clause-sum reduction | P(c) | Ch. 23 | Flat condition; Proposition 16.8; Berkowitz (I). |
| Truncation of constants and support of powers | P | Ch. 24 | Elementary. |
| Constants that are Laurent polynomials | I | Ch. 24 | Lemma N.2; coefficient step via Lemma M.1 not reproved. |
| Approximate sums with few terms | I | Ch. 24 | Lemma N.3. |
| Absence of approximate accelerators | I | Ch. 24 | Proposition N.4; machine construction verified in Lean only. |
| Border lower bound and closure of VP | I, P(c) | Ch. 24 | Theorem N.5; corollary derived here. |
| Approximate determinantal complexity | I, P(c) | Ch. 24 | Remark N.6; the one source statement with no Lean equivalent. |
| Accessibility profile and relations among levels | P | Ch. 25 | Definitions of Chapter 1; elementary. |
| Truth without computable threshold | P | Ch. 25 | Halting problem; self-contained. |
| Monotonicity of reconstruction cost in the environment | P | Ch. 25 | Definition of $C_K$. |
| Proof by cases without selection | P | Ch. 25 | Uses the alternative of Ch. 13 (I). |
| Explicit threshold for the flat case | P(c) | Ch. 25 | Appendix E: $m_0 = 2^{1600}$ relative to the inherited ingredients; contracting case open (Obligation 25.11). |
| Break-even against rediscovery and simplification | P | Ch. 26 | Elementary accounting model. |
| Properties of the net explanatory gain | P | Ch. 26 | Monotonicity under enrichment (Ch. 2). |
| Explanatory reach of spectral geometry | H | Ch. 26 | Conditional model; no evidence supplied. |
| Verification conditional on formalization; replay does not test fidelity | P | Ch. 27 | Elementary; definitions of Chapter 27. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Gate-count conventions differ by a factor of three | P | Ch. 27 | Elementary; the source records this factor as not formalized. |
| Lean definition of approximation equals the Laurent definition | P | Ch. 27 | Compared with the printed text of Statement.lean. |
| Lean verification of the four main propositions | I | Ch. 27 | Source Section 3 and Appendix Q; reported, not reproduced here (Obligations 27.11 and 27.12). |
| Non-identifiability and best rate of recovery | P | Ch. 28 | Counting; no assumption on the histories. |
| Effect of a provenance record | P | Ch. 28 | Refinement of fibers. |
| Provenance of the source proofs | H | Ch. 28 | Claim of the source; no evidence supplied here (Obligation 28.13). |
| Implementation identity is not instance identity; shared designation insufficient | P | Ch. 29 | Elementary; definitions of Chapter 29. |
| Continuation is a relation, not a function | P | Ch. 29 | Elementary. |
| Calculator principle | Pr | Ch. 29 | Normative proposal. |
| Monotone correction record; orphaned artifacts are frozen | P | Ch. 30 | Definitions of Chapter 30. |
| Responsibility is not validity | P | Ch. 30 | Independence of components. |
| Prospective authorship relation | Pr | Ch. 30 | Normative proposal (Obligation 30.9). |
| Delegation does not imply authorization; renaming invariance | P | Ch. 31 | Definitions of Chapter 31. |
| Responsibility follows control | Pr | Ch. 31 | Normative proposal (Obligation 31.7). |
| Importance does not imply urgency; publicity not determined by value | P | Ch. 32 | Elementary; definitions of Chapter 32. |
| Importance without urgency never justifies interruption | P | Ch. 33 | Definition of the surplus. |
| Contextual non-identifiability with regret bound | P | Ch. 33 | Linear in the interruption probability. |
| Interruption authority | Pr | Ch. 33 | Normative principle. |
| Constant pacing under convex expenditure, burst penalty, allocation | P | Ch. 34 | Hölder inequality; constant recovery assumed. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Interruption accumulation and output under interruption | P | Ch. 35 | Corollary of Chapter 34; restoration-proportional-to-loss is a hypothesis. |
| Feasible-set contraction | P | Ch. 35 | Elementary. |
| Empirical claims on urgency, interruption and dispossession | H | Ch. 32 to 35 | Conditional on the models (Obligation 35.9). |
| Local admissibility insufficient; rank one characterizes factorization | P | Ch. 36 | Elementary; flattening of the indicator. |
| Admissibility presheaf and gluing | P | Ch. 37 | Elementary; sheaf language. |
| Linear interface lower bound with equality | P | Ch. 37 | Rank factorization. |
| Rank bounds gluing expenditure | H | Ch. 37 | Needs an explicit reconstruction model (Hypothesis 37.7). |
| Fiber criterion, envelopes, monotonicity of descent | P | Ch. 38 | Elementary. |
| Boolean evaluation has maximal distortion | P | Ch. 38 | Proposition of Chapter 22; degree bound for circuits. |
| Dependency shadow: what descends and what does not | P | Ch. 38 | Elementary; classification and graph left abstract. |
| Network cut rank, interface feasibility, equality for wire networks | P | Ch. 39 | Chapter 37; cut-rank identity of Chapter 9. |
| Architecture and physical bandwidth interpretation | H | Ch. 39 | Model assumption (Obligation 39.8). |
| Obstruction certificate, resource lower bound, covering obstruction | P | Ch. 40 | Elementary. |
| Asymptotic completeness for tensors | I | Ch. 40 | Theorem A.6; abstract analogue not assumed. |
| Geometric obstruction principle | H | Ch. 40 | Research hypothesis; no new curvature theorem claimed. |
| Dependency graph, closure status, frontier, and coverage as a progress indicator | P | Ch. 41 | Elementary; statuses are conventions (Definition 41.2). |
| Finite computations certify only their instances | P | Ch. 41 | Euler's polynomial; elementary. |
| Export of the ledger as a graph | H | Ch. 41 | Open (Obligation 41.11). |
| Wire rank, path versus complete network, crossing wires | P | Ch. 42 | Cut-rank identity of Chapter 9; exact finite checks agree. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Permanent minus determinant is twice the odd terms; Boolean single-variable example; amortized average | P | Ch. 42 | Elementary. |
| Numerical residual and finite data do not decide identities or exponents | P | Ch. 43 | Elementary. |
| Midpoint margin vanishes in Euclidean space and is negative on the sphere | P | Ch. 43 | Explicit computation; trees checked numerically only. |
| Reproducibility record of the experiments of Chapter 43 | C | Ch. 43 | One environment; second environment open (Obligation 43.10). |
| Dependency invalidation and bound on loss of support | P | Ch. 44 | Elementary; invalidation is not falsity. |
| Independent maintenance is monotone and needs modification rights | P | Ch. 44 | Definitions of Chapter 44. |
| Repository layout and policy | H | Ch. 44 | Open (Obligation 44.9). |
| Polynomial time without practicality; hardness-based and information-theoretic security distinguished | P | Ch. 25 | Explicit numerical example (Example 25.10). |
| Recovery and determination along a reporting chain; verification does not determine evaluative attributes | P | Ch. 28 | Elementary; the automated summary is a dated excerpt of public representation only. |
| Certificate-conditioned reuse theorem | P | Ch. 45 | Combines Chapters 1, 2, 26; accounting model. |
| Conditional geometric lower bound and its flat-case instance | P | Ch. 46 | Arithmetic of the template; inputs (spectral inequality, gap) as in Chapters 15, 16. |
| Independence from discovery history; acceptance alone is not portability; portable translations compose | P | Ch. 47 | Elementary. |
| Enlargement of the continuation space; preservation is not implied by output quality | P | Ch. 48 | Elementary; definitions of Chapter 48. |

| Result | Status | Location | Dependencies and remarks |
|---|---|---|---|
| Amplification and not substitution | Pr | Ch. 48 | Normative proposal. |
| Deferral without loss, rational nonreaction, batching of deferred events | P | Ch. 49 | Elementary; Chapters 32, 33. |
| Translation of the source development to an independent environment | H | Ch. 47 | Open (Obligation 47.5). |
| Noncommutative, border-determinantal and valid-division bounds | I | Ch. 6 | Raz 2026, OpenAI 2026, Guo 2026; cited under their own models and not used in any proof here. |

# Appendix I

# Computational Experiments

## I.1 Scope

This appendix reproduces the code of the computations reported in Chapters 11 and 42 to 43, with the output of each, so that the computations can be rerun and compared. The first listing computes flattening ranks of graph tensors, the amortization examples, the margin of the midpoint inequality in three spaces, and the synthetic scaling comparison. The second listing is the exact finite check of the crossing identity. The reproducibility record of Definition 43.8 is in Chapter 43: the environment was Python 3.13 with SymPy 1.14.0 and NumPy 2.5.3, and the source of the crossing-identity check is the manuscript with the digest recorded there. The random number generators are seeded in the first listing, so that the figures reported in Chapter 43 are reproduced exactly by it.

## I.2 Experiments of Chapters 42 and 43

```
import itertools, math, random
import numpy as np, sympy as sp
from fractions import Fraction

def wire_tensor_graph(n, edges, q):
    """Dense tensor T_G; party v has index space prod over incident edges of [q]."""
    inc={v:[i for i,(a,b) in enumerate(edges) if v in (a,b)] for v in range(n)}
    dims=[q**len(inc[v]) for v in range(n)]
    return inc,dims

def flat_rank(n, edges, q, S):
    """Exact rank of flattening of T_G across S|S^c. Entries: 1 iff both endpoints agree on each
    ↪  edge index."""
    inc,dims=wire_tensor_graph(n,edges,q)
    Sc=[v for v in range(n) if v not in S]
    # a configuration of the whole edge-index vector determines a unique party configuration.
    rows={}; cols={}
    entries=[]
    for assign in itertools.product(range(q),repeat=len(edges)):
        def party_cfg(v):
            return tuple(assign[i] for i in inc[v])
```

```
        r=tuple(party_cfg(v) for v in S); c=tuple(party_cfg(v) for v in Sc)
        rows.setdefault(r,len(rows)); cols.setdefault(c,len(cols))
        entries.append((rows[r],cols[c]))
    M=np.zeros((len(rows),len(cols)),dtype=np.int64)
    for i,j in entries: M[i,j]=1
    # exact rank over Q via sympy would be slow; use fraction-free rank over a large prime and numpy
    ↪  float cross-check
    return M.shape, int(np.linalg.matrix_rank(M.astype(float))), rank_mod_p(M,2**31-1)

def rank_mod_p(M,p):
    A=M.copy()%p; r=0; rows,cols=A.shape
    for c in range(cols):
        piv=None
        for i in range(r,rows):
            if A[i,c]%p: piv=i;break
        if piv is None: continue
        A[[r,piv]]=A[[piv,r]]
        inv=pow(int(A[r,c]),-1,p)
        A[r]=(A[r]*inv)%p
        for i in range(rows):
            if i!=r and A[i,c]%p: A[i]=(A[i]-A[i,c]*A[r])%p
        r+=1
        if r==rows: break
    return r

print("== Example 42.1 wire")
W=sp.Matrix([[1,0],[0,1]]); print("rank W_2 =",W.rank())

print("== Example 42.2 path vs complete (q=2)")
for k,S in [(4,{0,1}),(5,{0,1}),(6,{0,1,2})]:
    pass
path=lambda k:[(i,i+1) for i in range(k-1)]
comp=lambda k:list(itertools.combinations(range(k),2))
for k,s in [(4,2),(6,3),(8,4)]:
    sh,r,rp=flat_rank(k,path(k),2,set(range(s)))
    print(f"path P_{k} |S|={s} shape {sh} rank(float)={r} rank(mod p)={rp} formula=2")
for k,s in [(4,2),(5,2),(5,1),(6,1)]:
    sh,r,rp=flat_rank(k,comp(k),2,set(range(s)))
    print(f"complete K_{k} |S|={s} shape {sh} rank(float)={r} rank(mod p)={rp}
    ↪  formula={2**(s*(k-s))}")
print("K_8 |S|=4 formula",2**16,"; direct matrix not built (2^28 rows)")

print("== Example 42.3 Boolean")
x=sp.symbols('x'); f=x**2-x; g=sp.Integer(0)
print("E(f)==E(g):",all(f.subs(x,v)==g for v in (0,1)),"; f==g:",sp.expand(f-g)==0)

print("== Example 42.4 perm vs det 2x2 and 3x3")
a,b,c,d=sp.symbols('a b c d')
M2=sp.Matrix([[a,b],[c,d]])
perm2=a*d+b*c; det2=M2.det()
print("perm-det =",sp.expand(perm2-det2))
def perm(M):
    n=M.shape[0]
    return sum(sp.prod([M[i,s[i]] for i in range(n)]) for s in itertools.permutations(range(n)))
syms=sp.symbols('m0:9'); M3=sp.Matrix(3,3,syms)
diff3=sp.expand(perm(M3)-M3.det())
print("3x3 perm-det has",len(diff3.as_ordered_terms()),"terms; all coefficients
↪  even:",all(int(co)%2==0 for co in sp.Poly(diff3,*syms).coeffs()))
# char 2 over GF(2) for random 0/1 matrices, n=4
```

```
random.seed(0); ok=True
for _ in range(200):
    n=4; A=sp.Matrix(n,n,lambda i,j: random.randint(0,1))
    if (perm(A)-A.det())%2!=0: ok=False
print("200 random 4x4 0/1 matrices: perm = det mod 2:",ok)

print("== Example 42.5 amortization")
D=Fraction(1000); a_=Fraction(1)
for N in (1,10,100,1000,10**6): print(N, float(a_+D/N))

print("== break-even example: d=11,a=1,F=1000,mu=4 -> N >",Fraction(1000)/(11-1-4))

print("== Exp 43: midpoint margin")
def margin(dzx,dzxp,dxxp,dzy): return 0.5*dzx**2+0.5*dzxp**2-0.25*dxxp**2-dzy**2
random.seed(1)
# Euclidean R^3
np.random.seed(1)
mn=1e9; mx=-1e9
for _ in range(100000):
    X=np.random.randn(4,3); xx,xp,z=X[0],X[1],X[2]; y=(xx+xp)/2
    m=margin(*[np.linalg.norm(u) for u in (z-xx,z-xp,xx-xp,z-y)]); mn=min(mn,m);mx=max(mx,m)
print("Euclid R^3 margin range:",mn,mx)
# tree: metric tree star with 3 legs
def tree_d(p,q):  # points (leg,t) t>=0, leg in 0..2
    (l1,t1),(l2,t2)=p,q
    return abs(t1-t2) if (l1==l2 or t1==0 or t2==0) else t1+t2
mn=1e9
for _ in range(200000):
    P=lambda:(random.randrange(3),random.random()*2)
    xx,xp,z=P(),P(),P()
    dxx=tree_d(xx,xp)
    # midpoint on geodesic
    if xx[0]==xp[0] or xx[1]==0 or xp[1]==0:
        if xx[0]==xp[0]: y=(xx[0],(xx[1]+xp[1])/2)
        else:
            l=xx[0] if xp[1]==0 else xp[0]; t=(xx[1]+xp[1])/2; y=(l,t)
            if xx[1]==0 or xp[1]==0:
                y=(xx[0] if xx[1]>0 else xp[0],dxx/2)
    else:
        h=dxx/2
        y=(xx[0],xx[1]-h) if xx[1]>=h else (xp[0],xp[1]-h)
        if y[1]<0: y=(0,0)
        y=(y[0],abs(y[1]))
    assert abs(tree_d(xx,y)-dxx/2)<1e-9 and abs(tree_d(y,xp)-dxx/2)<1e-9
    m=margin(tree_d(z,xx),tree_d(z,xp),dxx,tree_d(z,y)); mn=min(mn,m)
print("star tree margin min (>=0 expected):",mn)
# sphere S^2 great-circle distance, positive curvature
def sph(u,v): return math.acos(max(-1,min(1,float(np.dot(u,v)))))
np.random.seed(2)
neg=0;tot=0;mn=1e9
for _ in range(100000):
    U=np.random.randn(3,3); U/=np.linalg.norm(U,axis=1)[:,None]
    xx,xp,z=U
    if sph(xx,xp)>math.pi-0.2: continue   # skip nearly antipodal pairs
    y=xx+xp; y/=np.linalg.norm(y)
    m=margin(sph(z,xx),sph(z,xp),sph(xx,xp),sph(z,y))
    tot+=1; neg+= m< -1e-12; mn=min(mn,m)
print("sphere S^2: configurations",tot,"with negative margin",neg,"smallest margin",mn)
```

```
print("== Exp 43: finite data cannot fix the exponent")
lm=np.array([4+0.5*i for i in range(0,121)])  # log2 m from 4 to 64
data=lm**(4/3)
for al in (1.25,4/3,1.4):
    # best c in least squares of log-ratio
    c=np.exp(np.mean(np.log(data)-al*np.log(lm)))
    err=np.max(np.abs(np.log(c*lm**al)-np.log(data)))
    print(f"alpha={al:.4f} best c={c:.3f} max |log ratio|={err:.3f}  Q_alpha ratio last/first={
    ↪  (data[-1]/lm[-1]**al)/(data[0]/lm[0]**al):.3f}")
```

## I.3 Output of the First Listing

```
== Example 42.1 wire
rank W_2 = 2
== Example 42.2 path vs complete (q=2)
path P_4 |S|=2 shape (4, 4) rank(float)=2 rank(mod p)=2 formula=2
path P_6 |S|=3 shape (8, 8) rank(float)=2 rank(mod p)=2 formula=2
path P_8 |S|=4 shape (16, 16) rank(float)=2 rank(mod p)=2 formula=2
complete K_4 |S|=2 shape (32, 32) rank(float)=16 rank(mod p)=16 formula=16
complete K_5 |S|=2 shape (128, 512) rank(float)=64 rank(mod p)=64 formula=64
complete K_5 |S|=1 shape (16, 1024) rank(float)=16 rank(mod p)=16 formula=16
complete K_6 |S|=1 shape (32, 32768) rank(float)=32 rank(mod p)=32 formula=32
K_8 |S|=4 formula 65536 ; direct matrix not built (2^28 rows)
== Example 42.3 Boolean
E(f)==E(g): True ; f==g: False
== Example 42.4 perm vs det 2x2 and 3x3
perm-det = 2*b*c
3x3 perm-det has 3 terms; all coefficients even: True
200 random 4x4 0/1 matrices: perm = det mod 2: True
== Example 42.5 amortization
1 1001.0
10 101.0
100 11.0
1000 2.0
1000000 1.001
== break-even example: d=11,a=1,F=1000,mu=4 -> N > 500/3
== Exp 43: midpoint margin
Euclid R^3 margin range: -1.4210854715202004e-14 1.7763568394002505e-14
star tree margin min (>=0 expected): -5.329070518200751e-15
sphere S^2: configurations 99040 with negative margin 99040 smallest margin -8.644745358748159
== Exp 43: finite data cannot fix the exponent
alpha=1.2500 best c=1.321 max |log ratio|=0.163  Q_alpha ratio last/first=1.260
alpha=1.3333 best c=1.000 max |log ratio|=0.000  Q_alpha ratio last/first=1.000
alpha=1.4000 best c=0.800 max |log ratio|=0.130  Q_alpha ratio last/first=0.831
```

## I.4 Crossing Identity Check

The second listing evaluates determinants of explicit matrices built for the crossing identity on small cases with exact polynomial arithmetic and tests the degree bound, the reflection symmetry, the support conditions, the number and values of the nonzero entries, and the rank bound. The rank-bound part holds with room to spare in these cases, so it does not discriminate, as the ledger records.

```
import itertools, sympy as sp
X,t=sp.symbols('X t'); y=X**2
def Bc(r,n):
    s=sp.series((1+t)**r/(1+y*t)**(r+1),t,0,n+1).removeO()
    return sp.expand(s.coeff(t,n))
def Cc(r,n):
    s=sp.series((1+y*t)**r/(1+t)**(r+1),t,0,n+1).removeO()
    return sp.expand(s.coeff(t,n))
def symbols_of(size,energy,maxel=8):
    out=[]
    for I in itertools.combinations(range(maxel),size):
        for J in itertools.combinations(range(maxel),size):
            if sum(2*i+1 for i in I)+sum(2*j+1 for j in J)==energy: out.append((I,J))
    return out
def Pcount(E):
    return sum(len(symbols_of(n,e)) for n in range(0,E+1) for e in range(0,E+1) if e<=E) if False
    ↪  else sum(len(symbols_of(n,e)) for n in range(0,E+1) for e in range(0,E+1))
Bm={};Cm={}
def B(r,n):
    if (r,n) not in Bm: Bm[(r,n)]=Bc(r,n)
    return Bm[(r,n)]
def C(r,n):
    if (r,n) not in Cm: Cm[(r,n)]=Cc(r,n)
    return Cm[(r,n)]
def build(sa,sb,sc,sd):
    # rows: I_a,I_d,I_c,I_b ; columns: J_a,J_d,J_c,J_b
    rows=[('a',i) for i in sa[0]]+[('d',i) for i in sd[0]]+[('c',i) for i in sc[0]]+[('b',i) for i
    ↪  in sb[0]]
    cols=[('a',j) for j in sa[1]]+[('d',j) for j in sd[1]]+[('c',j) for j in sc[1]]+[('b',j) for j
    ↪  in sb[1]]
    n=len(rows); M=sp.zeros(n,n)
    for r,(pr,i) in enumerate(rows):
        for c,(pc,j) in enumerate(cols):
            d=lambda a: 1 if a else 0
            e=d(i==j)
            tab={('a','a'):0,('a','d'):C(i,j),('a','c'):e*X**(2*i+1),('a','b'):(y-1)*X**(2*i+1)*B(i
            ↪  ,j),
                 ('d','a'):-C(j,i),('d','d'):0,('d','c'):X**(2*j+1)*B(j,i),('d','b'):-e*X**(2*i+1),
                 ('c','a'):-e*X**(2*i+1),('c','d'):-X**(2*i+1)*B(i,j),('c','c'):0,('c','b'):-(y-1)*
                 ↪  C(i,j),
                 ('b','a'):-(y-1)*X**(2*j+1)*B(j,i),('b','d'):e*X**(2*i+1),('b','c'):(y-1)*C(j,i),(
                 ↪  'b','b'):0}
            M[r,c]=tab[(pr,pc)]
    return M
def Psi(sa,sb,sc,sd):
    M=build(sa,sb,sc,sd)
    return sp.expand(M.det()) if M.shape[0]>0 else sp.Integer(1)
def run(n1,e1,n2,e2):
    S1=symbols_of(n1,e1); S2=symbols_of(n2,e2); Th=2*(e1+e2)
    print("case",(n1,e1,n2,e2),"|S1|",len(S1),"|S2|",len(S2),"Theta",Th)
    cache={}
    ok_deg=ok_refl=True; vals1={}
    for sa in S1:
     for sc in S1:
      for sb in S2:
       for sd in S2:
        p=Psi(sa,sb,sc,sd); cache[(sa,sb,sc,sd)]=p
        if sp.degree(p,X)>Th if p!=0 else False: ok_deg=False
```

```
        vals1[(sa,sb,sc,sd)]=p.subs(X,1)
    for (sa,sb,sc,sd),p in cache.items():
        lhs=sp.expand(X**Th*p.subs(X,1/X)); rhs=cache[(sa,sd,sc,sb)] if (sa,sd,sc,sb) in cache else
        ↪ None
        # reflection exchanges b and d: indices of b and d live in S2 both, a,c in S1, so valid
        if rhs is not None and sp.expand(lhs-rhs)!=0: ok_refl=False
    nz=[(k,v) for k,v in vals1.items() if v!=0]
    cond=all((k[2][1]==k[0][0] and k[2][0]==k[0][1] and k[1][1]==k[3][0] and k[1][0]==k[3][1]) for
    ↪ k,v in nz)
    signs=set(v for k,v in nz)
    expected=sum(1 for sa in S1 for sb in S2)
    print(" deg<=Theta:",ok_deg," reflection:",ok_refl," X=1 supported on conditions:",cond,"
    ↪ #nonzero:",len(nz),"expected",expected," value set:",signs)
    # rank check (b): coefficient of X^N as matrix rows (sa,sd) cols (sb,sc)
    for N in range(0,Th+1):
        rows=[(sa,sd) for sa in S1 for sd in S2]; cols=[(sb,sc) for sb in S2 for sc in S1]
        M=sp.Matrix(len(rows),len(cols),lambda r,c:
        ↪ sp.Poly(cache[(rows[r][0],cols[c][0],cols[c][1],rows[r][1])],X).coeff_monomial(X**N) if
        ↪ cache[(rows[r][0],cols[c][0],cols[c][1],rows[r][1])]!=0 else 0)
        rk=M.rank(); bound=(N+1)*Pcount(N)
        if rk>bound: print("  RANK VIOLATION at N",N,rk,bound)
    print(" rank bound (b) checked for N=0..Theta")
for case in [(1,2,1,2),(1,4,1,2),(1,4,1,4),(2,6,1,2)]:
    run(*case)
```

## I.5 Output of the Second Listing

```
case (1, 2, 1, 2) |S1| 1 |S2| 1 Theta 8
 deg<=Theta: True  reflection: True  X=1 supported on conditions: True  #nonzero: 1 expected 1
 ↪ value set: {1}
 rank bound (b) checked for N=0..Theta
case (1, 4, 1, 2) |S1| 2 |S2| 1 Theta 12
 deg<=Theta: True  reflection: True  X=1 supported on conditions: True  #nonzero: 2 expected 2
 ↪ value set: {1}
 rank bound (b) checked for N=0..Theta
case (1, 4, 1, 4) |S1| 2 |S2| 2 Theta 16
 deg<=Theta: True  reflection: True  X=1 supported on conditions: True  #nonzero: 4 expected 4
 ↪ value set: {1}
 rank bound (b) checked for N=0..Theta
case (2, 6, 1, 2) |S1| 0 |S2| 1 Theta 16
 deg<=Theta: True  reflection: True  X=1 supported on conditions: True  #nonzero: 0 expected 0
 ↪ value set: set()
 rank bound (b) checked for N=0..Theta
```

# Appendix J

# Correction History

## J.1 Principle

The correction record of an artifact is monotone, as Proposition 30.4 states: a correction adds to the record and removes nothing. This appendix is the record of corrections made to the statements of this monograph while it was developed, including corrections to statements of the planning outline on which it is based. Each entry gives the location, the error, the repair and the effect on status. Typographical and typesetting repairs are listed together at the end. No inherited result was changed by any correction.

## J.2 Corrections to Statements

| Location | Error | Repair | Effect on status |
|---|---|---|---|
| Chapters 22 to 24 | Circuit size was first counted with variables and constants free, contradicting the convention of Chapter 3 and of the source, under which they count | Definitions and constants restated, with $L_B \le 4s + n + 2$, determinantal transfer $N \ge (s/(C + 5V))^{1/c_1}$ and entry cost at most $4V + 1$; the factor-three translation proved in Chapter 27 (Proposition 27.9) | Statements revised; none inherited |

| Location | Error | Repair | Effect on status |
|---|---|---|---|
| Chapter 34 and outline Proposition on pacing | Hölder bound stated without the path constraint on capacity | Constant recovery assumed and the constraint $E(t) \geq E_{\min}$ for all $t$ stated; admissibility of constant effort proved (Proposition 34.2) | Statement revised |
| Chapter 26 and outline definitions of break-even and gain | Defined on the untruncated knowledge environment, which can be infinite | Defined on the truncated environment $K^{(M)}$ | Statement revised |
| Chapter 36, outline Proposition on local admissibility | Strict inclusion of the global set in the product of its coordinate projections was asserted as general | Inclusion always holds, with equality if and only if the set is a product (Proposition 36.2) | Statement revised |
| Chapter 38, outline Corollary on exact preservation | Values restricted to the reals, excluding infinite upper envelopes | Values taken in the extended reals (Definition 38.1) | Statement revised |
| Chapter 40, outline Theorem on the resource lower bound | Did not require $\varphi(y) > 0$ and defined the inverse of $g$ without an infimum | Hypothesis $\varphi(y) > 0$ added and $g^{-1}$ defined as an infimum (Theorem 40.5) | Statement revised |
| Chapter 45 | Third part of the reuse theorem was stated with a convergence condition not needed for the limit superior | Restated as a limit superior with convergence when the mean converges (Theorem 45.1) | Statement revised |

| Location | Error | Repair | Effect on status |
|---|---|---|---|
| Chapters 9 and 25 and the ledger | Spectral exponents were listed as inherited although their proof uses only the axioms of Chapter 8; the flat-case threshold was listed as a hypothesis although it follows from the inequalities of the proof | Proved in Appendix C (Proposition C.2) and Appendix E (Proposition E.1) | Ledger rows updated; downstream inherited items unchanged |
| Chapter 43 and the experiment script | The script lacked fixed seeds and the exclusion of nearly antipodal pairs that the text states, so reported figures could not be regenerated exactly | Seeds and the exclusion made explicit; figures regenerated and confirmed against the text (Appendix I) | Record added |
| Chapter 6 and bibliography | References to the later works of Raz, OpenAI and Guo were absent from an earlier draft | Restored at the request of the author, with their model restrictions preserved and with the statement that no proof here uses them | Ledger row restored |

## J.3 Typographical and Typesetting Repairs

A missing label for the arithmetization proposition and undefined theorem environments for applications and examples were supplied. A formula in the proposition on the absence of a determinantal analogue was corrected from a malformed product of the clause parameters to the product $k\ell$. A line in the proof of the principal bound that overflowed because of a long commit identifier was repaired by a global setting that allows stretching of inter-word space, which also removed an earlier overflow in a theorem proof of Part VII. None of these changed a statement.

# Bibliography


[1] M. Agrawal and V. Vinay, Arithmetic Circuits: A Chasm at Depth Four, in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008, 67–75.

[2] M. Agrawal, N. Kayal, and N. Saxena, PRIMES Is in P, *Annals of Mathematics* 160 (2004), 781–793.

[3] A. Alder, Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht, Dissertation, Universität Zürich, 1984.

[4] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.

[5] P. Bürgisser, The Complexity of Factors of Multivariate Polynomials, *Foundations of Computational Mathematics* 4 (2004), 369–396.

[6] W. Baur and V. Strassen, The Complexity of Partial Derivatives, *Theoretical Computer Science* 22 (1983), 317–330.

[7] S. J. Berkowitz, On Computing the Determinant in Small Parallel Time Using a Small Number of Processors, *Information Processing Letters* 18 (1984), 147–150.

[8] M. R. Bridson and A. Haefliger, *Metric Spaces of Non-Positive Curvature*, Springer, 1999.

[9] F. Bruhat and J. Tits, Groupes réductifs sur un corps local: I. Données radicielles valuées, *Publications Mathématiques de l'IHÉS* 41 (1972), 5–251.

[10] P. Bürgisser, *Completeness and Reduction in Algebraic Complexity Theory*, Springer, 2000.

[11] P. Bürgisser, Cook's Versus Valiant's Hypothesis, *Theoretical Computer Science* 235 (2000), 71–88.

[12] S. A. Cook and R. A. Reckhow, The Relative Efficiency of Propositional Proof Systems, *Journal of Symbolic Logic* 44 (1979), 36–50.

[13] M. A. Forbes, Low-Depth Algebraic Circuit Lower Bounds over Any Field, in *39th Computational Complexity Conference*, LIPIcs 300, 2024, 31:1–31:16.

[14] D. G. Glynn, The Permanent of a Square Matrix, *European Journal of Combinatorics* 31 (2010), 1887–1891.

[15] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, Arithmetic Circuits: A Chasm at Depth 3, *SIAM Journal on Computing* 45 (2016), 1064–1079.

[16] Discussion thread on the VP versus VNP announcement, subreddit r/ClaudeAI, including the stickied automated summary by the forum's moderation bot, retrieved November 24, 2026. `https://www.reddit.com/r/ClaudeAI/comments/1y3w563/`

[17] J. Guo, An $n^2 \log\log n$ Lower Bound for Permanent Circuits with Valid Division, arXiv:2609.29568, version 2, October 8, 2026.

[18] A. Haken, The Intractability of Resolution, *Theoretical Computer Science* 39 (1985), 297–308.

[19] L. Hyafil, On the Parallel Evaluation of Multivariate Polynomials, *SIAM Journal on Computing* 8 (1979), 120–123.

[20] M. Jerrum and M. Snir, Some Exact Complexity Results for Straight-Line Computations over Semirings, *Journal of the ACM* 29 (1982), 874–897.

[21] P. Koiran, Arithmetic Circuits: The Chasm at Depth Four Gets Wider, *Theoretical Computer Science* 448 (2012), 56–65.

[22] N. Limaye, S. Srinivasan, and S. Tavenas, Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits, *Journal of the ACM* 72 (2025), article 26.

[23] K. D. Mulmuley and M. Sohoni, Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems, *SIAM Journal on Computing* 31 (2001), 496–526.

[24] K. D. Mulmuley and M. Sohoni, Geometric Complexity Theory II: Towards Explicit Obstructions for Embeddings among Class Varieties, *SIAM Journal on Computing* 38 (2008), 1175–1206.

[25] N. Nisan, Lower Bounds for Non-Commutative Computation, in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, 1991, 410–418.

[26] OpenAI, A Cubic Lower Bound for Border Determinantal Complexity of the Permanent, research manuscript with accompanying Lean formalization, September 24, 2026.

[27] R. Raz, Polynomial Lower Bounds for Arithmetic Circuits over Non-Commutative Rings, ECCC TR26-061 and arXiv:2604.22006, 2026.

[28] R. Raz, Multi-Linear Formulas for Permanent and Determinant Are of Super-Polynomial Size, *Journal of the ACM* 56 (2009), article 8.

[29] H. J. Ryser, *Combinatorial Mathematics*, Carus Mathematical Monographs 14, Mathematical Association of America, 1963.

[30] Claude (Anthropic), VP $\neq$ VNP over Every Field, manuscript (`paper.pdf`) with accompanying Lean formalization, repository `anthropics/formal-math`, directory `vp-vnp`, commit `918c8e2778aee18b0283dc5a33ed39ab3cd63a96` (commit message "vp-vnp: VP != VNP, paper and Lean proof"), November 23, 2026; the manuscript states that the development uses Lean 4.33.1 and Mathlib at commit `0df444a360eaa60ab8c11dca51a86af692955474`. `https://github.com/anthropics/formal-math/commit/918c8e2778aee18b0283dc5a33ed39ab3cd63a96`

[31] V. Strassen, Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten, *Numerische Mathematik* 20 (1973), 238–251.

[32] V. Strassen, The Asymptotic Spectrum of Tensors, *Journal für die reine und angewandte Mathematik* 384 (1988), 102–152.

[33] S. Tavenas, Improved Bounds for Reduction to Depth 4 and Depth 3, *Information and Computation* 240 (2015), 2–11.

[34] L. G. Valiant, Completeness Classes in Algebra, in *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, 1979, 249–261.

[35] L. G. Valiant, The Complexity of Computing the Permanent, *Theoretical Computer Science* 8 (1979), 189–201.

[36] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff, Fast Parallel Computation of Polynomials Using Few Processors, *SIAM Journal on Computing* 12 (1983), 641–644.

[37] A. Yehudayoff, Separating Monotone VP and VNP, in *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, 2019, 425–429.