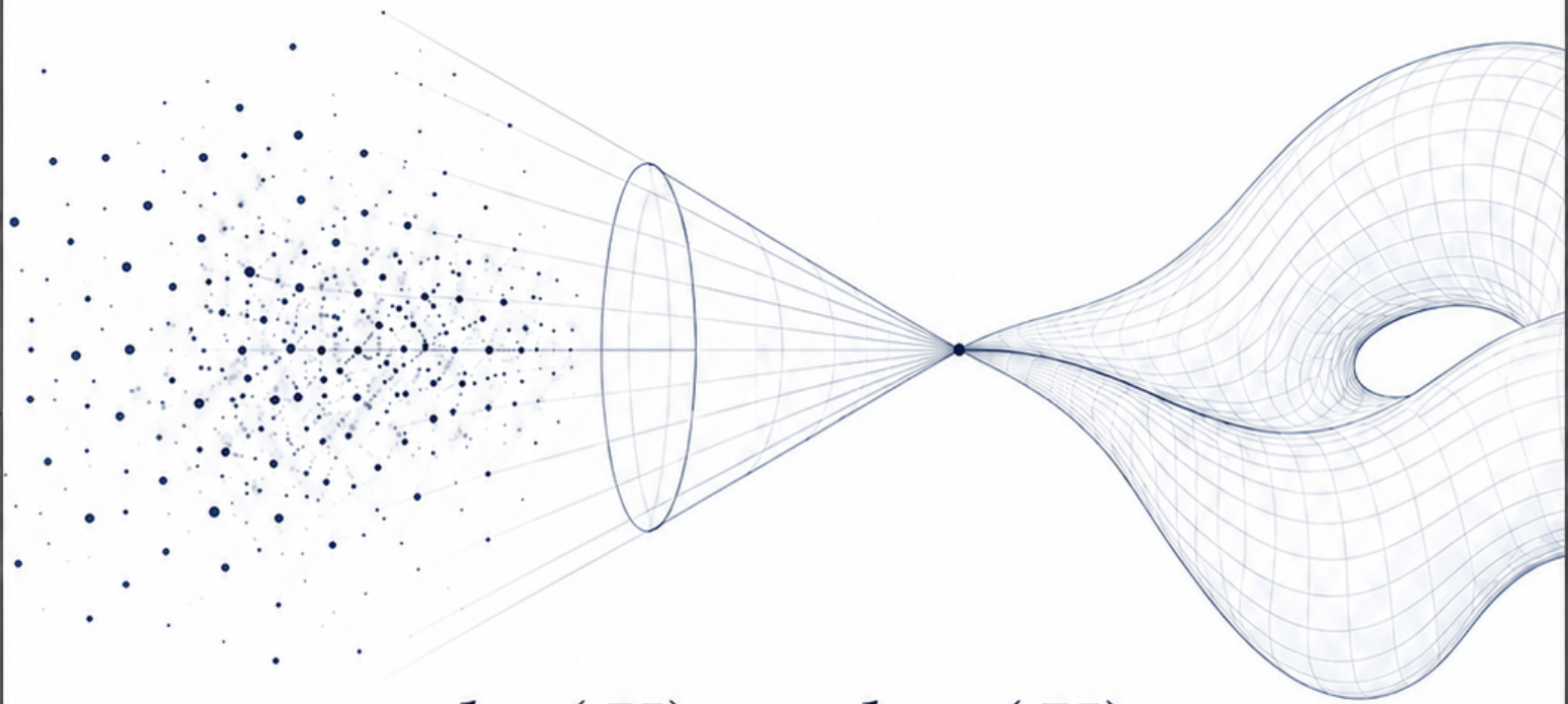


DRCC

DIMENSIONAL REDUCTION VIA CONTROLLED COMBINATORICS



$$d_{\text{rec}}(X) \leq d_{\text{frag}}(X)$$

DRCC CRITERION

*DRCC is proposed as a toolbox for controlled reduction,
designed to identify structural principles implicitly present
in successful mathematical solution strategies.*

— DRCC-V2-REW2 —

Reza Hesamiy

Munich, Germany

2026

Contents

9	Full Adder, Dadda Multiplier, and Structured Digital Systems	2
9.1	Purpose	2
9.2	Full-Adder Model	2
9.3	DRCC Collapse by Hamming Weight	2
9.4	Output Reconstruction from Structural Classes	3
9.5	Reconstruction Space	3
9.6	Controlled Reconstruction Operator	4
9.7	Complete Finite Verification	4
9.8	Runtime Count	4
9.9	From Full Adders to Dadda Multipliers and Digital Systems	5
9.10	Transition Interpretation	7
9.11	Interpretation	7
9.12	Scope, Limitations, and Summary	7
10	Empirical Validation Protocol and Benchmark Baselines	8
10.1	Purpose and Scope	8
10.2	Validation Objects and Quantities	8
10.3	Benchmark Baselines and Fairness	9
10.4	Reproducible Validation Protocol	9
10.5	Finite Structural Validation Records	10
10.6	Acceptance and Transition Reporting	11
10.7	Scope, Non-Claims, and Conclusion	11
11	Cross-Case Numerical Evidence and Baseline Comparison	12
11.1	Purpose and Scope	12
11.2	Common Quantities and Evidence Levels	12
11.3	Worked Count-Level Example: Housing	13
11.4	Consolidated Cross-Case Evidence	13
11.5	Finite Criterion, Limitations, and Conclusion	14
12	Relation to Established Literature and Methods	15
12.1	Scope and Comparison Interface	15
12.2	Search, Propagation, and Problem-Specific Solvers	16
12.3	Data Reduction, Symmetry, and Structural Width	16
12.4	Tensor Representation and Digital Arithmetic	17
12.5	Consolidated Comparison and Evidence Obligations	17
12.6	Novelty Boundary and Conclusion	18
13	Scope, Limitations, Open Problems, and Non-Claims	18
13.1	Scope	18
13.2	Formal Limitations	19
13.3	Empirical and Algorithmic Limitations	20
13.4	Open Problems	20
13.5	Non-Claims	21
A	DRCC Formula, Symbol, and Verification Register	22
A.1	Core Symbol Register	22
A.2	Core Formula Register	22
A.3	Evidence and Acceptance Levels	23
A.4	Case-Study and OPA Register	23
A.5	Boundary and Reference Notes	24

9 Full Adder, Dadda Multiplier, and Structured Digital Systems

9.1 Purpose

This chapter gives a complete finite DRCC micro-case and then shows how its local arithmetic invariant extends to structured digital systems. The development is

$$\text{full adder} \longrightarrow \text{Dadda multiplier} \longrightarrow \text{structured digital systems.} \quad (9.1)$$

The full-adder part is exact and completely enumerable. It specifies the classical input space, the controlled collapse map, the structural classes, the reconstruction spaces, the output-reconstruction operator, and a count-level cost model. The Dadda part is an established hardware architecture used here only as a larger example of value-preserving compression followed by final reconstruction. Neither example is used to infer a universal speedup or an asymptotic complexity result.

9.2 Full-Adder Model

Let

$$\mathbb{B} = \{0, 1\} \quad (9.2)$$

be the Boolean alphabet, and let P_{FA} denote the full-adder problem instance. Its classical input space and output space are

$$X_{\text{FA}} = \mathbb{B}^3, \quad Y_{\text{FA}} = \mathbb{B}^2. \quad (9.3)$$

An input is written as

$$x = (a, b, c_{\text{in}}) \in X_{\text{FA}}, \quad (9.4)$$

where a , b , and c_{in} are Boolean bits. The full-adder map is

$$F_{\text{FA}} : X_{\text{FA}} \longrightarrow Y_{\text{FA}}, \quad (9.5)$$

with

$$F_{\text{FA}}(a, b, c_{\text{in}}) = (s, c_{\text{out}}), \quad (9.6)$$

where \oplus denotes exclusive OR and \vee denotes Boolean OR:

$$\begin{aligned} s &= a \oplus b \oplus c_{\text{in}}, \\ c_{\text{out}} &= (ab) \vee (ac_{\text{in}}) \vee (bc_{\text{in}}). \end{aligned} \quad (9.7)$$

The fragmentation dimension is therefore

$$d_{\text{frag}}(P_{\text{FA}}) = |X_{\text{FA}}| = 2^3 = 8. \quad (9.8)$$

9.3 DRCC Collapse by Hamming Weight

Definition 9.1 (Full-Adder Collapse Map). Define the reduced structural space by

$$Z_{\text{FA}} = \{0, 1, 2, 3\}, \quad (9.9)$$

and define

$$C_{\text{FA}} : X_{\text{FA}} \longrightarrow Z_{\text{FA}} \quad (9.10)$$

by

$$C_{\text{FA}}(a, b, c_{\text{in}}) = a + b + c_{\text{in}}. \quad (9.11)$$

Thus, $C_{\text{FA}}(x)$ is the Hamming weight of x .

For each $k \in Z_{\text{FA}}$, define the structural class

$$\mathcal{C}_k = \{x \in X_{\text{FA}} : C_{\text{FA}}(x) = k\}. \quad (9.12)$$

The four classes are

$$\begin{aligned} \mathcal{C}_0 &= \{(0, 0, 0)\}, \\ \mathcal{C}_1 &= \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}, \\ \mathcal{C}_2 &= \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}, \\ \mathcal{C}_3 &= \{(1, 1, 1)\}. \end{aligned} \quad (9.13)$$

Let

$$\mathfrak{C}_{\text{FA}} = \{\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\} \quad (9.14)$$

be the family of collapse fibers. Then

$$|X_{\text{FA}}| = 8, \quad |Z_{\text{FA}}| = |\mathfrak{C}_{\text{FA}}| = 4, \quad (9.15)$$

so the finite structural collapse is

$$\mathbb{B}^3 \longrightarrow \{0, 1, 2, 3\}, \quad 8 \longrightarrow 4. \quad (9.16)$$

9.4 Output Reconstruction from Structural Classes

Define the class-output map

$$\widehat{F}_{\text{FA}} : \mathfrak{C}_{\text{FA}} \longrightarrow Y_{\text{FA}} \quad (9.17)$$

by

$$\begin{aligned} \widehat{F}_{\text{FA}}(\mathcal{C}_0) &= (0, 0), & \widehat{F}_{\text{FA}}(\mathcal{C}_1) &= (1, 0), \\ \widehat{F}_{\text{FA}}(\mathcal{C}_2) &= (0, 1), & \widehat{F}_{\text{FA}}(\mathcal{C}_3) &= (1, 1). \end{aligned} \quad (9.18)$$

Hence the output is constant on every structural class. In particular, if $x, y \in X_{\text{FA}}$ satisfy $C_{\text{FA}}(x) = C_{\text{FA}}(y)$, then

$$F_{\text{FA}}(x) = F_{\text{FA}}(y). \quad (9.19)$$

Because the four classes in Equation (9.18) produce four distinct elements of Y_{FA} , no exact factorization of F_{FA} through fewer than four reachable intermediate states is possible. Thus, Z_{FA} is a minimal output-sufficient state space.

This is output sufficiency, not input invertibility. The three distinct inputs in \mathcal{C}_1 , for example, have the same output but cannot be recovered uniquely from the class value.

9.5 Reconstruction Space

For $k \in Z_{\text{FA}}$, define the reconstruction space by

$$\Omega_{P_{\text{FA}}}(k) = \{x \in X_{\text{FA}} : C_{\text{FA}}(x) = k\}. \quad (9.20)$$

Thus,

$$\Omega_{P_{\text{FA}}}(k) = \mathcal{C}_k. \quad (9.21)$$

The reconstruction dimension is

$$d_{\text{rec}}(P_{\text{FA}}, k) = |\Omega_{P_{\text{FA}}}(k)|, \quad (9.22)$$

and its four values are

$$(d_{\text{rec}}(P_{\text{FA}}, 0), d_{\text{rec}}(P_{\text{FA}}, 1), d_{\text{rec}}(P_{\text{FA}}, 2), d_{\text{rec}}(P_{\text{FA}}, 3)) = (1, 3, 3, 1). \quad (9.23)$$

Consequently, for every $k \in Z_{\text{FA}}$,

$$0 < d_{\text{rec}}(P_{\text{FA}}, k) \leq d_{\text{frag}}(P_{\text{FA}}) = 8 < \infty. \quad (9.24)$$

The set $\Omega_{P_{\text{FA}}}(k)$ reconstructs the original input fiber. It is conceptually distinct from the output-reconstruction operator defined next.

9.6 Controlled Reconstruction Operator

Define

$$Q_{\text{FA}} : Z_{\text{FA}} \longrightarrow Y_{\text{FA}} \quad (9.25)$$

by

$$Q_{\text{FA}}(k) = \left(k \bmod 2, \left\lfloor \frac{k}{2} \right\rfloor \right). \quad (9.26)$$

Because $k \in \{0, 1, 2, 3\}$, the second coordinate in Equation (9.26) is Boolean.

Theorem 9.2 (Exact Full-Adder Factorization). *For every $x \in X_{\text{FA}}$,*

$$F_{\text{FA}}(x) = Q_{\text{FA}}(C_{\text{FA}}(x)). \quad (9.27)$$

Equivalently,

$$F_{\text{FA}} = Q_{\text{FA}} \circ C_{\text{FA}}. \quad (9.28)$$

Proof. Let $x = (a, b, c_{\text{in}}) \in X_{\text{FA}}$ and set $k = C_{\text{FA}}(x) = a + b + c_{\text{in}}$. The parity of k is the sum bit, and $\lfloor k/2 \rfloor$ is one exactly when at least two input bits are active. Therefore $Q_{\text{FA}}(k) = (s, c_{\text{out}}) = F_{\text{FA}}(x)$. \square

The complete output-preserving chain is therefore

$$\mathbb{B}^3 \xrightarrow{C_{\text{FA}}} \{0, 1, 2, 3\} \xrightarrow{Q_{\text{FA}}} \mathbb{B}^2. \quad (9.29)$$

9.7 Complete Finite Verification

Table 9.1: Full-adder inputs, structural classes, and reconstructed outputs.

a	b	c_{in}	k	s	c_{out}	Fiber
0	0	0	0	0	0	\mathcal{C}_0
0	0	1	1	1	0	\mathcal{C}_1
0	1	0	1	1	0	\mathcal{C}_1
1	0	0	1	1	0	\mathcal{C}_1
0	1	1	2	0	1	\mathcal{C}_2
1	0	1	2	0	1	\mathcal{C}_2
1	1	0	2	0	1	\mathcal{C}_2
1	1	1	3	1	1	\mathcal{C}_3

Table 9.1 verifies all eight inputs and all four reduced states. The four DRCC acceptance conditions are compactly resolved as follows: structural consistency follows from Theorem 9.2; the reconstruction bound and finiteness follow from Equation (9.24); and controlled output reconstruction follows from the exact factorization in Equation (9.28). A runtime claim requires the additional accounting given in Section 9.8.

9.8 Runtime Count

The structural state-space gain is defined by

$$G_{\text{FA,struct}} = \frac{|X_{\text{FA}}|}{|Z_{\text{FA}}|}. \quad (9.30)$$

Hence,

$$G_{\text{FA,struct}} = \frac{8}{4} = 2. \quad (9.31)$$

This quantity measures quotient-state compression only. To define a count-level implementation comparison, let

$$c_C, c_Q, c_D \in \mathbb{N}_{>0} \quad (9.32)$$

be, respectively, the elementary-operation counts for computing C_{FA} , applying Q_{FA} , and evaluating a chosen direct full-adder baseline. Then

$$\begin{aligned} N_{\text{Collapse}}(P_{\text{FA}}) &= c_C, \\ N_{\text{Reconstruction}}(P_{\text{FA}}, k) &= c_Q, \\ N_{\text{DRCC}}(P_{\text{FA}}, k) &= c_C + c_Q, \\ N_{\text{Classic}}(P_{\text{FA}}) &= c_D. \end{aligned} \quad (9.33)$$

The corresponding count-level gain is

$$G_{\text{FA,count}}(k) = \frac{c_D}{c_C + c_Q}. \quad (9.34)$$

A favorable count-level regime requires

$$c_C + c_Q < c_D, \quad (9.35)$$

or equivalently $G_{\text{FA,count}}(k) > 1$. For wall-clock measurement, let t_C , t_Q , and t_D be the measured times of the same three procedures under one fixed platform and measurement protocol. Then

$$G_{\text{FA,time}}(k) = \frac{t_D}{t_C + t_Q}, \quad t_C, t_Q, t_D \in \mathbb{R}_{>0}. \quad (9.36)$$

Neither Equation (9.35) nor a wall-clock gain follows from $G_{\text{FA,struct}} = 2$. The structural, count-level, and measured quantities are distinct.

9.9 From Full Adders to Dadda Multipliers and Digital Systems

9.9.1 Coupled Full-Adders

Let $n \in \mathbb{N}_{>0}$ and define

$$I_n = \{0, 1, \dots, n-1\}. \quad (9.37)$$

For a ripple-carry adder, the local input of cell $i \in I_n$ is $(a_i, b_i, c_i) \in \mathbb{B}^3$. Its reduced state and output are

$$k_i = a_i + b_i + c_i, \quad (s_i, c_{i+1}) = Q_{\text{FA}}(k_i). \quad (9.38)$$

Define the coordinate projections

$$\pi_1, \pi_2 : \mathbb{B}^2 \longrightarrow \mathbb{B}, \quad \pi_1(s, c) = s, \quad \pi_2(s, c) = c. \quad (9.39)$$

Then

$$s_i = \pi_1(Q_{\text{FA}}(k_i)), \quad c_{i+1} = \pi_2(Q_{\text{FA}}(k_i)). \quad (9.40)$$

Thus, local full-adder reconstructions are exact but not independent; the carry output of one cell is an input of the next cell.

9.9.2 Partial-Product Structure

For $m \in \mathbb{N}_{>0}$, write an m -bit vector least-significant bit first as $u = (u_0, \dots, u_{m-1}) \in \mathbb{B}^m$, and define its unsigned value by

$$\text{val}_m(u) = \sum_{r=0}^{m-1} 2^r u_r. \quad (9.41)$$

The map val_m is a bijection from \mathbb{B}^m to $\{0, 1, \dots, 2^m - 1\}$. Denote its inverse by

$$\text{bin}_m : \{0, 1, \dots, 2^m - 1\} \longrightarrow \mathbb{B}^m. \quad (9.42)$$

For operands $A = (A_i)_{i \in I_n} \in \mathbb{B}^n$ and $B = (B_j)_{j \in I_n} \in \mathbb{B}^n$, define the partial-product bits by

$$p_{ij} = A_i B_j, \quad (i, j) \in I_n \times I_n. \quad (9.43)$$

Let

$$X_{\text{PP}}(n) = \mathbb{B}^{I_n \times I_n} \quad (9.44)$$

be the space of binary partial-product matrices, and define the generation map

$$P_n : \mathbb{B}^n \times \mathbb{B}^n \longrightarrow X_{\text{PP}}(n), \quad P_n(A, B) = (A_i B_j)_{(i,j) \in I_n^2}. \quad (9.45)$$

For $R = (r_{ij}) \in X_{\text{PP}}(n)$, define its weighted value by

$$\text{wval}_n(R) = \sum_{i \in I_n} \sum_{j \in I_n} 2^{i+j} r_{ij}. \quad (9.46)$$

For every operand pair (A, B) ,

$$\text{wval}_n(P_n(A, B)) = \text{val}_n(A) \text{val}_n(B). \quad (9.47)$$

9.9.3 Dadda Compression and Final Reconstruction

A full adder applied to three bits $x, y, z \in \mathbb{B}$ in a column of weight 2^ℓ , where $\ell \in \mathbb{N}_0$, returns $(s, c) = Q_{\text{FA}}(x + y + z)$ and satisfies

$$2^\ell(x + y + z) = 2^\ell s + 2^{\ell+1} c. \quad (9.48)$$

The same identity with $z = 0$ gives the half-adder case. A Dadda reduction network is a specified composition of such local counters that reduces the partial-product matrix to two binary rows while preserving Equation (9.48) at every stage.

Define the valid two-row state space by

$$Z_{\text{DAD}}(n) = \{(u, v) \in \mathbb{B}^{2n} \times \mathbb{B}^{2n} : \text{val}_{2n}(u) + \text{val}_{2n}(v) < 2^{2n}\}. \quad (9.49)$$

Let

$$C_{\text{DAD},n} : X_{\text{PP}}(n) \longrightarrow Z_{\text{DAD}}(n) \quad (9.50)$$

be the deterministic map induced by one fixed Dadda reduction network. If

$$C_{\text{DAD},n}(R) = (u_R, v_R), \quad (9.51)$$

then the defining arithmetic invariant is

$$\text{wval}_n(R) = \text{val}_{2n}(u_R) + \text{val}_{2n}(v_R). \quad (9.52)$$

Let

$$Y_{\text{prod}}(n) = \mathbb{B}^{2n} \quad (9.53)$$

be the product-output space. The final-adder reconstruction map is

$$Q_{\text{DAD},n} : Z_{\text{DAD}}(n) \longrightarrow Y_{\text{prod}}(n) \quad (9.54)$$

with

$$Q_{\text{DAD},n}(u, v) = \text{bin}_{2n}(\text{val}_{2n}(u) + \text{val}_{2n}(v)). \quad (9.55)$$

Finally, define the unsigned multiplication map

$$M_n : \mathbb{B}^n \times \mathbb{B}^n \longrightarrow Y_{\text{prod}}(n) \quad (9.56)$$

by

$$M_n(A, B) = \text{bin}_{2n}(\text{val}_n(A) \text{val}_n(B)). \quad (9.57)$$

Proposition 9.3 (Dadda Value-Preserving Factorization). *For the specified Dadda reduction network,*

$$M_n = Q_{\text{DAD},n} \circ C_{\text{DAD},n} \circ P_n. \quad (9.58)$$

Proof. For $(A, B) \in \mathbb{B}^n \times \mathbb{B}^n$, Equation (9.47) gives the weighted value of the generated partial-product matrix. Equation (9.52) preserves that value in the two compressed rows, and Equation (9.55) converts their sum to the unique $2n$ -bit product. This is exactly Equation (9.57). \square

The architecture in Proposition 9.3 is an established Dadda multiplier, not a DRCC invention. Its relevance is methodological: a structured intermediate representation is reduced by local value-preserving operations and reconstructed by a final adder. Actual engineering comparisons require specified baselines and measured delay, area, power, and energy; those measurements are not supplied by the factorization alone.

9.10 Transition Interpretation

The full adder has one fixed input size. Its relation $8 \rightarrow 4$ is therefore a finite structural threshold, not an asymptotic transition theorem. It proves that four output-relevant states suffice for this local task; it does not define a family-wide runtime crossover.

For a parameterized implementation family indexed by $n \in \mathbb{N}_{>0}$, a transition point may be reported only after both baseline and reduced runtimes have been specified. If n_w is the first tested size at which the reduced implementation is faster and $R_w \in \mathbb{R}_{>0}$ is its measured runtime in seconds, then

$$W = (n_w, R_w[s]). \quad (9.59)$$

No value of W is inferred in this chapter, because no common implementation, hardware platform, or measured runtime series is fixed.

9.11 Interpretation

The full-adder result identifies the minimal state needed to reconstruct its output: the Hamming weight is sufficient, whereas the original input identity is generally not recoverable. The Dadda factorization uses the same local arithmetic invariant at a larger scale, but its compressed state is a pair of value-preserving rows rather than a four-element quotient.

The common DRCC-compatible pattern is controlled intermediate reduction with exact output reconstruction. The mathematical form of that pattern is reusable; its operation count, circuit depth, area, power, and wall-clock consequences remain implementation-dependent.

9.12 Scope, Limitations, and Summary

Table 9.2: Established results and explicit boundaries of Chapter 9.

Category	Statement
Established	The full-adder map factors exactly through four Hamming-weight states; its input fibers have sizes 1, 3, 3, 1; and $G_{\text{FA,struct}} = 2$.
Established	A specified Dadda network preserves the weighted partial-product value and reconstructs the exact $2n$ -bit product through a final adder.
Not established	The structural reduction does not by itself prove a lower operation count, a wall-clock speedup, an asymptotic gain, or polynomial-time solvability.
Non-claim	DRCC does not invent the full adder or the Dadda multiplier, and this finite hardware analysis does not imply a general complexity collapse.

Chapter 9 therefore supplies one exact finite DRCC micro-case and one carefully bounded bridge to established digital arithmetic. The next chapter may test such structures only through an explicit benchmark, baseline, measurement, and reproducibility protocol.

10 Empirical Validation Protocol and Benchmark Baselines

10.1 Purpose and Scope

This chapter defines one common protocol for evaluating DRCC reductions. Its purpose is to connect structural analysis to reproducible count-level and wall-clock comparisons without identifying those evidence levels. For every benchmark, the protocol requires an explicit candidate space, a controlled reduction, an admissible reconstruction rule, a stated baseline, complete cost accounting, and a correctness check.

A finite structural reduction is evidence about representation and reconstruction. A count-level gain is evidence under a stated work-unit model. A wall-clock gain is an empirical result on a specified implementation and platform. Persistence over a tested instance family is a scaling observation. None of these statements alone is an asymptotic complexity theorem.

10.2 Validation Objects and Quantities

Let \mathcal{P} be a class of finite problem instances and let $P \in \mathcal{P}$. Denote the finite classical candidate space by $X(P)$, and let

$$X_{\text{adm}}(P) \subseteq X(P) \quad (10.1)$$

be the set of candidates satisfying the benchmark's stated admissibility or solution criterion. Let $Z(P)$ be a finite reduced structural space and let

$$C_P : X(P) \longrightarrow Z(P) \quad (10.2)$$

be the controlled reduction map. For a reduced state $\zeta \in Z(P)$, define the associated reconstruction space by

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \quad (10.3)$$

The fragmentation and reconstruction dimensions are

$$d_{\text{frag}}(P) = |X(P)|, \quad d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|. \quad (10.4)$$

Whenever $d_{\text{rec}}(P, \zeta) > 0$, the reconstruction-space reduction factor is

$$\rho_{\text{rec}}(P, \zeta) = \frac{d_{\text{frag}}(P)}{d_{\text{rec}}(P, \zeta)}. \quad (10.5)$$

This ratio is structural. It is not a runtime gain.

Let $N_{\text{Classic}}(P)$ be the number of declared elementary work units used by the selected classical baseline. For exhaustive enumeration with $c_{\text{eval}}(P) \in \mathbb{N}_{>0}$ work units per candidate,

$$N_{\text{Classic}}(P) = d_{\text{frag}}(P)c_{\text{eval}}(P). \quad (10.6)$$

Let $N_{\text{Collapse}}(P)$ count construction of the reduced structure and let $N_{\text{Reconstruction}}(P, \zeta)$ count admissible reconstruction and evaluation. The complete DRCC count is

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta), \quad (10.7)$$

and the count-level gain is

$$G_{\text{count}}(P, \zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P, \zeta)}. \quad (10.8)$$

For measured execution, let $T_{\text{Classic}}(P)$, $T_{\text{Collapse}}(P)$, and $T_{\text{Reconstruction}}(P, \zeta)$ denote wall-clock times in seconds. Then

$$T_{\text{DRCC}}(P, \zeta) = T_{\text{Collapse}}(P) + T_{\text{Reconstruction}}(P, \zeta), \quad (10.9)$$

and the empirical time gain is

$$G_{\text{time}}(P, \zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P, \zeta)}. \quad (10.10)$$

Peak memory, denoted by $M_{\text{Classic}}(P)$ and $M_{\text{DRCC}}(P, \zeta)$, is reported separately and is not absorbed into either gain.

Proposition 10.1 (Count–Time Equivalence under a Common-Rate Model). *Assume that a common effective rate $f \in \mathbb{R}_{>0}$ converts the declared work units of both procedures into seconds, with no omitted setup, communication, or memory cost. If*

$$T_{\text{Classic}}(P) = \frac{N_{\text{Classic}}(P)}{f}, \quad T_{\text{DRCC}}(P, \zeta) = \frac{N_{\text{DRCC}}(P, \zeta)}{f}, \quad (10.11)$$

then

$$G_{\text{time}}(P, \zeta) = G_{\text{count}}(P, \zeta), \quad (10.12)$$

and

$$T_{\text{DRCC}}(P, \zeta) < T_{\text{Classic}}(P) \iff N_{\text{DRCC}}(P, \zeta) < N_{\text{Classic}}(P). \quad (10.13)$$

Proof. Both runtimes are divided by the same positive number f . Division by f preserves the strict inequality, and cancellation of f in the runtime ratio gives Equation (10.12). \square

The proposition is a model statement. It does not authorize replacing measured time by counts when work units, memory behavior, or platforms differ.

10.3 Benchmark Baselines and Fairness

Let $\mathcal{B} \subseteq \mathcal{P}$ be a benchmark family and let \mathcal{O} be its declared output-report space. A baseline implementation and a DRCC implementation are maps

$$A_{\text{base}}, A_{\text{DRCC}} : \mathcal{B} \longrightarrow \mathcal{O}. \quad (10.14)$$

For every tested $P \in \mathcal{B}$, correctness requires

$$A_{\text{DRCC}}(P) = A_{\text{base}}(P), \quad (10.15)$$

where equality means the same decision, admissible output set, or objective value according to the benchmark contract.

Exhaustive enumeration is an appropriate reference baseline for a finite micro-case. A performance claim for a non-trivial family must also use a relevant established domain-specific baseline. The comparison is accepted only under the following common conditions.

Table 10.1: Fair-comparison conditions for a DRCC benchmark.

Condition	Required information
Same task	Identical instance, admissibility criterion, objective, and required output.
Explicit methods	Complete baseline algorithm, collapse map, and reconstruction procedure.
Complete costs	Parsing, preprocessing, collapse, reconstruction, verification, and postprocessing are included.
No hidden oracle	Neither method receives information unavailable to the other method.
Comparable execution	Same platform and comparable implementation effort, or an explicit explanation of differences.
Correctness	Equation (10.15) is verified before performance is interpreted.

10.4 Reproducible Validation Protocol

For each benchmark family, the validation procedure is applied once in the following order.

- V1.** Specify \mathcal{B} , the size parameter, instance generator, random seeds, and the candidate and output spaces.
- V2.** Specify A_{base} , C_P , the construction of $Z(P)$, the selected state or state rule, and the reconstruction procedure for $\Omega_P(\zeta)$.
- V3.** Verify output agreement before comparing counts or times.
- V4.** Record all count-level quantities in Equations (10.4)–(10.8).

- V5.** On one stated platform, record collapse, reconstruction, total runtime, baseline runtime, and peak memory over a declared repetition protocol.
- V6.** Report count-level, wall-clock, and scaling conclusions in separate fields; report a transition only for a parameterized family.

The minimum reproducibility record is compactly summarized below.

Table 10.2: Minimum reproducibility record.

Field	Required record
Instances	Family, sizes, generator or source, parameters, seeds, and exclusions.
Platform	Processor or hardware target, memory, operating system, compiler or language, and versions.
Implementations	Baseline method, DRCC collapse and reconstruction, data structures, and build settings.
Measurements	Counts, component times, total times, peak memory, number of runs, aggregation, and dispersion.
Verification	Output agreement, failed instances, timeouts, and handling of infeasible states.
Availability	Source code, scripts, input data, and commands sufficient to reproduce the tables.

An illustrative count is not described as an empirical experiment unless an implementation, platform, repetitions, and measured times are available.

10.5 Finite Structural Validation Records

The following two records show how exact finite structures enter the protocol. They verify structural quantities and correctness; they do not supply wall-clock evidence.

10.5.1 Graph 3-Coloring on the Four-Cycle

Let P_{C_4} be graph 3-coloring on the cycle with

$$V = \{v_1, v_2, v_3, v_4\}, \quad E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\}, \quad K = \{1, 2, 3\}. \quad (10.16)$$

A coloring is a map $c : V \rightarrow K$, and the raw classical space is

$$X(P_{C_4}) = K^V, \quad d_{\text{frag}}(P_{C_4}) = 3^4 = 81. \quad (10.17)$$

Fix the fragment $\eta = (c(v_1), c(v_2)) = (1, 2)$. Its admissible reconstruction space is

$$\Omega_{C_4}(\eta) = \{c \in K^V : c(v_1) = 1, c(v_2) = 2, c(u) \neq c(v) \text{ for every } \{u, v\} \in E\}. \quad (10.18)$$

Direct enumeration gives

$$\Omega_{C_4}(\eta) = \{(1, 2, 1, 2), (1, 2, 1, 3), (1, 2, 3, 2)\}, \quad (10.19)$$

where each tuple is ordered as (v_1, v_2, v_3, v_4) . Hence

$$d_{\text{rec}}(P_{C_4}, \eta) = 3, \quad \rho_{\text{rec}}(P_{C_4}, \eta) = \frac{81}{3} = 27. \quad (10.20)$$

This result is fragment-specific. It does not set $N_{\text{DRCC}} = 3$, because the costs of selecting the fragment, constructing constraints, and producing the three completions must still be counted.

10.5.2 Full-Adder Output Reconstruction

Let $\mathbb{B} = \{0, 1\}$, let $X_{\text{FA}} = \mathbb{B}^3$, and let $Z_{\text{FA}} = \{0, 1, 2, 3\}$. The collapse map is

$$C_{\text{FA}} : X_{\text{FA}} \longrightarrow Z_{\text{FA}}, \quad C_{\text{FA}}(a, b, c_{\text{in}}) = a + b + c_{\text{in}}. \quad (10.21)$$

Let $F_{\text{FA}} : X_{\text{FA}} \rightarrow \mathbb{B}^2$ be the full-adder output map and define

$$Q_{\text{FA}} : Z_{\text{FA}} \longrightarrow \mathbb{B}^2, \quad Q_{\text{FA}}(k) = \left(k \bmod 2, \left\lfloor \frac{k}{2} \right\rfloor \right). \quad (10.22)$$

The exact output-reconstruction identity is

$$F_{\text{FA}} = Q_{\text{FA}} \circ C_{\text{FA}}. \quad (10.23)$$

Moreover,

$$|X_{\text{FA}}| = 8, \quad |Z_{\text{FA}}| = 4, \quad (|C_{\text{FA}}^{-1}(k)|)_{k=0}^3 = (1, 3, 3, 1). \quad (10.24)$$

Thus the output-relevant state ratio is

$$\rho_{\text{FA, state}} = \frac{|X_{\text{FA}}|}{|Z_{\text{FA}}|} = 2. \quad (10.25)$$

This is an exact structural result, consistent with Chapter 9. It is not a measured gate-delay or wall-clock speedup and is not equal to the size of every individual reconstruction fiber.

10.6 Acceptance and Transition Reporting

A report must identify the level of claim being accepted. Structural acceptance requires

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \quad (10.26)$$

A strict structural reduction additionally requires $d_{\text{rec}}(P, \zeta) < d_{\text{frag}}(P)$. Count-level advantage requires

$$N_{\text{DRCC}}(P, \zeta) < N_{\text{Classic}}(P) \iff G_{\text{count}}(P, \zeta) > 1. \quad (10.27)$$

Empirical runtime advantage requires output agreement, the fairness conditions in Table 10.1, and

$$T_{\text{DRCC}}(P, \zeta) < T_{\text{Classic}}(P) \iff G_{\text{time}}(P, \zeta) > 1. \quad (10.28)$$

For a parameterized benchmark family $(P_n)_{n \in I}$, where $I \subseteq \mathbb{N}_{>0}$ is an ordered set of tested sizes, define

$$G_{\text{time}}(n) = \frac{T_{\text{Classic}}(P_n)}{T_{\text{DRCC}}(P_n, \zeta_n)}. \quad (10.29)$$

If an exact crossing occurs at $n_w \in I$, define

$$G_{\text{time}}(n_w) = 1, \quad W = (n_w, R_w[s]), \quad R_w = T_{\text{Classic}}(P_{n_w}) = T_{\text{DRCC}}(P_{n_w}, \zeta_{n_w}). \quad (10.30)$$

If no exact equality occurs, report the nearest tested crossing interval

$$W_{\text{int}} = [n_-, n_+], \quad G_{\text{time}}(n_-) < 1 < G_{\text{time}}(n_+). \quad (10.31)$$

No transition is inferred from one isolated instance. Persistence on a finite tested range is reported as empirical scaling evidence, not as a worst-case or asymptotic theorem.

10.7 Scope, Non-Claims, and Conclusion

Chapter 10 supplies one benchmark definition, one cost decomposition, one fairness rule set, one reproducibility record, and one acceptance scheme. The finite C_4 and full-adder records demonstrate exact structural quantities, but they are not empirical runtime experiments.

The protocol does not claim that every controlled collapse is useful, that every smaller reconstruction space yields a runtime gain, that a toy example establishes scaling, that tensorization is a solver, or that DRCC proves polynomial-time solvability or $P = NP$. A DRCC performance claim is accepted only for the stated benchmark family and tested range for which correctness, complete collapse and reconstruction costs, fair baselines, and reproducible measurements have been supplied.

The resulting methodological chain is

$$\begin{aligned} \text{benchmark} &\longrightarrow \text{baseline and DRCC procedures} \longrightarrow \text{correctness} \\ &\longrightarrow \text{counts and measurements} \longrightarrow \text{level-specific conclusion.} \end{aligned} \quad (10.32)$$

This separation prevents structural compression, finite count gain, wall-clock improvement, observed scaling, and asymptotic complexity from being reported as interchangeable results.

11 Cross-Case Numerical Evidence and Baseline Comparison

11.1 Purpose and Scope

This chapter consolidates the finite case records used in the manuscript. It does not repeat the detailed Full-Adder derivation of Chapter 9 or the validation protocol of Chapter 10. Instead, it distinguishes three kinds of evidence: structural reduction, finite count-level advantage, and measured wall-clock advantage.

Let \mathcal{P}_{fin} denote the finite problem instances or finite instance families considered here, and let $P \in \mathcal{P}_{\text{fin}}$. The classical candidate space is $X(P)$, the admissible candidate set is $X_{\text{adm}}(P) \subseteq X(P)$, and a controlled reduction is a map

$$C_P : X(P) \longrightarrow Z(P), \quad (11.1)$$

where $Z(P)$ is the reduced structural space. For $\zeta \in Z(P)$, the associated reconstruction space is

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \quad (11.2)$$

Thus, every case is reported through the common chain

$$X(P) \xrightarrow{C_P} Z(P) \longrightarrow \Omega_P(\zeta). \quad (11.3)$$

The examples instantiate this chain in different ways; they do not define one universal DRCC algorithm.

11.2 Common Quantities and Evidence Levels

The fragmentation dimension, structural-state count, and reconstruction dimension are, respectively,

$$\begin{aligned} d_{\text{frag}}(P) &= |X(P)|, \\ R_{\text{struct}}(P) &= |Z(P)|, \\ d_{\text{rec}}(P, \zeta) &= |\Omega_P(\zeta)|. \end{aligned} \quad (11.4)$$

Whenever the denominators are positive, two purely structural ratios are

$$K_{\text{class}}(P) = \frac{d_{\text{frag}}(P)}{R_{\text{struct}}(P)}, \quad K_{\text{fiber}}(P, \zeta) = \frac{d_{\text{frag}}(P)}{d_{\text{rec}}(P, \zeta)}. \quad (11.5)$$

Neither ratio is, by itself, a runtime gain. Let $N_{\text{Collapse}}(P)$ count the declared elementary operations used to construct or apply the reduction, and let $N_{\text{Reconstruction}}(P, \zeta)$ count the declared elementary operations used after reduction. The total DRCC count is

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \quad (11.6)$$

For a positive classical count $N_{\text{Classic}}(P)$ and a positive DRCC count, the count-level gain is

$$G_{\text{count}}(P, \zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P, \zeta)}. \quad (11.7)$$

If measured runtimes $T_{\text{Classic}}(P) > 0$ and $T_{\text{DRCC}}(P, \zeta) > 0$ are available, the wall-clock gain is

$$G_{\text{time}}(P, \zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P, \zeta)}. \quad (11.8)$$

Definition 11.1 (Evidence Levels). A case has *structural evidence* (level S) when the spaces, map, and relevant cardinalities are explicit. It has *count-level evidence* (level C) when, in addition, both terms in Equation 11.6 and the classical count are specified under one cost model. It has *empirical evidence* (level E) only when reproducible wall-clock measurements against a stated baseline are reported.

The levels are cumulative: E requires the information of C, and C requires the structural information of S.

11.3 Worked Count-Level Example: Housing

Let P_H be a finite housing-selection instance with candidate set

$$H = \{h_1, \dots, h_{1200}\}, \quad X(P_H) = H. \quad (11.9)$$

Assume that structural screening retains a set $H_{\text{red}} \subseteq H$ with $|H_{\text{red}}| = 25$. Let one screening operation cost $c_{\text{screen}} = 7$ units and one detailed evaluation cost $c_{\text{detail}} = 1000$ units. The classical procedure applies both operations to every candidate, so

$$N_{\text{Classic}}(P_H) = 1200(c_{\text{screen}} + c_{\text{detail}}) = 1,208,400. \quad (11.10)$$

The DRCC model screens all candidates and applies detailed evaluation only to the retained set. Hence,

$$\begin{aligned} N_{\text{Collapse}}(P_H) &= 1200c_{\text{screen}} = 8,400, \\ N_{\text{Reconstruction}}(P_H, \zeta_H) &= 25c_{\text{detail}} = 25,000, \\ N_{\text{DRCC}}(P_H, \zeta_H) &= 33,400, \end{aligned} \quad (11.11)$$

where ζ_H denotes the reduced state selecting H_{red} . The resulting finite count-level gain is

$$G_{\text{count}}(P_H, \zeta_H) = \frac{1,208,400}{33,400} = \frac{6042}{167} \approx 36.18. \quad (11.12)$$

This is a level-C result under the stated unit-cost model. It is not a measured wall-clock result, and it does not imply a universal speedup for filtering tasks.

11.4 Consolidated Cross-Case Evidence

The Full-Adder and four-cycle records are the finite structures verified in Chapters 9 and 10. For the SAT record, the concrete formula is

$$\Phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3). \quad (11.13)$$

Its eight assignments produce four clause-output vectors, and the satisfying vector $(1, 1, 1)$ has a fiber of size five. For the CSP record, let $D = \{1, 2, 3\}$ be the common domain of four variables. The reported value $d_{\text{rec}} = 9$ is retained only as a conditional record because the current source does not list the local constraint tables that produce those nine assignments.

For the coupled 16-bit adder, let $\mathbb{B} = \{0, 1\}$, and define

$$\nu_{16}(u) = \sum_{i=0}^{15} 2^i u_i, \quad u_{16} : \mathbb{B}^{16} \longrightarrow \{0, \dots, 2^{16} - 1\}. \quad (11.14)$$

The external-input/output map is

$$F_{16} : \mathbb{B}^{16} \times \mathbb{B}^{16} \times \mathbb{B} \longrightarrow \{0, \dots, 2^{17} - 1\}, \quad F_{16}(A, B, c) = \nu_{16}(A) + \nu_{16}(B) + c. \quad (11.15)$$

Its domain has 2^{33} states and its image has 2^{17} states. The internal carry bits are generated recursively and are therefore not independent external inputs.

Table 11.1: Cross-case evidence ledger. A dash means that the current manuscript does not specify the quantity required for a numerical claim.

Case	Classical record	Reduced record	Reonstr. record	Total DRCC count	Reported result	Evidence level, baseline, and scope
Full Adder	$d_{\text{frag}} = 8$ input states.	$ Z = 4$ Hamming-weight classes.	Input-fiber sizes 1, 3, 3, 1; one output pair is recovered from each class.	$c_C + c_Q$, where c_C and c_Q are the declared collapse and output-reconstruction costs.	$K_{\text{class}} = 2$; numerical G_{count} not fixed.	S. Baseline: direct Full-Adder evaluation. Exact, finite, and output-preserving; not input-invertible.

Continued on next page

Table 11.1 continued

Case	Classical record	Reduced record	Reconstr. record	Total DRCC count	Reported result	Evidence level, baseline, and scope
Graph 3-Coloring on C_4	$d_{\text{frag}} = 3^4 = 81$ raw colorings.	One fixed fragment: $c(v_1) = 1$, $c(v_2) = 2$.	$d_{\text{rec}} = 3$ admissible completions.	–; fragment selection, construction, and checking costs are not counted.	$K_{\text{fiber}} = 81/3 = 27$; not a runtime gain.	S. Baseline: enumeration or backtracking. Applies only to the stated fragment.
Three-variable SAT formula Φ	$2^3 = 8$ assignments.	Four reachable clause-output vectors.	Satisfying fiber size 5.	–; clause-vector construction and reconstruction costs are not reported.	$K_{\text{class}} = 8/4 = 2$; no count-level gain established.	S. Baseline: exhaustive evaluation for this instance; DPLL/CDCL for empirical solver comparison.
Four-variable CSP record	$ D ^4 = 3^4 = 81$ assignments.	Local admissible tables; total reduced-state count not reported.	Conditional assumption $d_{\text{rec}} = 9$.	–; table-construction and reconstruction costs are absent.	Conditional $K_{\text{fiber}} = 9$; no G_{count} .	Conditional S. Baseline: backtracking and propagation. Reproducibility requires the actual tables.
Six-city symmetric routing	$(6-1)!/2 = 60$ tours after fixing a start and identifying reversals.	Assumed 18 route classes.	Assumed at most 3 representatives per class; aggregate budget at most 54.	Complete count includes collapse and reconstruction; 54 alone is insufficient.	No numerical gain is valid without the collapse count. If reconstruction uses exactly 54 units, favorability requires $N_{\text{Collapse}} < 6$.	Conditional S. Baseline: exhaustive tours, a specified dynamic program, or branch-and-bound.
Housing instance	$N_{\text{Classic}} = 1,208,400$.	25 candidates retained from 1200.	Detailed count: 25,000.	$8,400 + 25,000 = 33,400$.	$G_{\text{count}} \approx 36.18$.	C. Baseline uses the same screening and detailed-evaluation costs. No wall-clock claim.
Coupled 16-bit adder	2^{33} external input triples (A, B, c) .	2^{17} reachable output sums.	Output fibers are class-dependent and not counted here.	–.	$K_{\text{class}} = 2^{16} = 65,536$; no runtime gain.	S. Baseline: an explicit adder implementation. Internal carries are coupled, so 8^{16} is not an external-input count.
Dadda multiplier family	For bit width n , 2^{2n} operand pairs; partial products are induced by the operands.	Value-preserving compressed column state; cardinality not reported.	Final carry-propagate addition reconstructs the product.	–.	Exact factorization and weighted-value preservation; no numerical gain assigned.	S. Baselines: established Dadda, Wallace, and related designs; required metrics are delay, area, power, and correctness.

Table 11.1 deliberately leaves cells blank when a collapse cost, reconstruction procedure, or empirical baseline is missing. Replacing such blanks by reconstruction-space cardinalities would conflate structural compression with a total evaluation count.

Remark 11.2 (Coupled Digital Systems). The local Full-Adder quotient $8 \rightarrow 4$ does not automatically exponentiate to a count-level speedup for a multi-bit adder. Internal carries constrain which local triples can coexist. Similarly, an n -bit multiplier has 2^{2n} external operand pairs; its n^2 partial-product bits are not independent candidate variables. Hardware gain requires an actual circuit comparison.

11.5 Finite Criterion, Limitations, and Conclusion

The final column of Table 11.1 states the minimum baseline obligation for each case. Chapter 10 supplies the common fairness and reproducibility protocol; repeating that protocol separately for every row would add prose without adding evidence.

Proposition 11.3 (Finite Count-Level DRCC Criterion). *Let $P \in \mathcal{P}_{\text{fin}}$, let $\zeta \in Z(P)$, and assume $N_{\text{Classic}}(P) > 0$ and $N_{\text{DRCC}}(P, \zeta) > 0$. Then*

$$G_{\text{count}}(P, \zeta) > 1 \iff N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta) < N_{\text{Classic}}(P). \quad (11.16)$$

Proof. Substitute Equation 11.6 into Equation 11.7. Since the two total counts are positive, the quotient is greater than one exactly when its numerator is greater than its denominator. \square

The proposition is an accounting equivalence, not a complexity theorem. The structural admissibility condition

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty \quad (11.17)$$

ensures a finite nonempty reconstruction space, but it does not determine the cost of obtaining or searching that space. Consequently,

$$K_{\text{class}}(P) > 1 \not\Rightarrow G_{\text{count}}(P, \zeta) > 1, \quad G_{\text{count}}(P, \zeta) > 1 \not\Rightarrow G_{\text{time}}(P, \zeta) > 1. \quad (11.18)$$

Among the numerical records summarized here, the housing model reaches level C because its collapse and reconstruction counts are both explicit. The Full Adder, four-cycle coloring, SAT, coupled adder, and Dadda records establish structural facts. The CSP and six-city routing numbers remain conditional until their construction procedures are fully specified. None of these records reaches level E without reproducible measured implementations.

The chapter therefore supports one precise finite conclusion:

A DRCC case is numerically favorable under a stated finite count model only
when admissible reconstruction is preserved and the complete
collapse-plus-reconstruction count is smaller than the classical count.

(11.19)

This conclusion neither supplies a universal solver nor establishes an asymptotic complexity result. In particular, it does not prove a general speedup for SAT, CSP, graph coloring, routing, or digital hardware, and it does not imply either $P = NP$ or $P \neq NP$.

12 Relation to Established Literature and Methods

12.1 Scope and Comparison Interface

This chapter positions DRCC relative to established methods for finite combinatorial search, constraint reasoning, preprocessing, structural decomposition, tensor analysis, and digital arithmetic. The relation is one of possible composition and comparison, not identity or replacement.

Let \mathcal{P}_{fin} be a class of finite problem instances and let $P \in \mathcal{P}_{\text{fin}}$. The classical candidate space is $X(P)$, the admissible candidate set is $X_{\text{adm}}(P) \subseteq X(P)$, and the reduced structural space is $Z(P)$. A controlled reduction is a map

$$C_P : X(P) \longrightarrow Z(P). \quad (12.1)$$

For a reduced state $\zeta \in Z(P)$, the associated reconstruction space is

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \quad (12.2)$$

The fragmentation and reconstruction dimensions are therefore

$$d_{\text{frag}}(P) = |X(P)|, \quad d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|. \quad (12.3)$$

Let A denote a specified baseline algorithm for P , and let $N_A(P) > 0$ be its evaluation count under a declared elementary-operation model. Let $N_{\text{Collapse}}(P)$ count construction or application of Equation (12.1), and let $N_{\text{Reconstruction}}(P, \zeta)$ count the declared work performed after reduction. Then

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta), \quad G_{\text{DRCC}|A}(P, \zeta) = \frac{N_A(P)}{N_{\text{DRCC}}(P, \zeta)}. \quad (12.4)$$

Proposition 12.1 (Method-Specific Count Criterion). *Assume that $N_A(P) > 0$ and $N_{\text{DRCC}}(P, \zeta) > 0$. Then DRCC has a finite count-level advantage over the specified baseline A if and only if*

$$N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta) < N_A(P). \quad (12.5)$$

Proof. By Equation (12.4) and positivity of both counts, $G_{\text{DRCC}|A}(P, \zeta) > 1$ holds exactly when $N_A(P) > N_{\text{DRCC}}(P, \zeta)$. Substitution of the defining sum for N_{DRCC} gives Equation (12.5). \square

A structural resemblance between DRCC and an established method does not supply this inequality. The baseline, reduction procedure, reconstruction procedure, and operation model must all be fixed before a numerical comparison is meaningful.

12.2 Search, Propagation, and Problem-Specific Solvers

Backtracking incrementally extends a partial candidate and rejects an extension when it cannot lead to an admissible solution [6]. Branch-and-bound supplements such a search tree with bounds that exclude subproblems that cannot improve the incumbent solution [11]. These are algorithmic traversal principles. DRCC instead specifies the map in Equation (12.1) and the fiber in Equation (12.2). A DRCC implementation may use backtracking or branch-and-bound to construct or search a fiber, but their visited nodes and bounding operations must then be included in $N_{\text{Reconstruction}}$.

Constraint propagation, SAT solving, and graph coloring share a common finite predicate representation. Define the Boolean alphabet $\mathbb{B} = \{0, 1\}$. Let $\mathbb{N} = \{1, 2, \dots\}$, and suppose that P has $m \in \mathbb{N}$ constraint predicates

$$\kappa_j : X(P) \longrightarrow \mathbb{B}, \quad j \in \{1, \dots, m\}. \quad (12.6)$$

Their constraint-signature map is

$$\sigma_P : X(P) \longrightarrow \mathbb{B}^m, \quad \sigma_P(x) = (\kappa_1(x), \dots, \kappa_m(x)). \quad (12.7)$$

Writing $\mathbf{1}_m = (1, \dots, 1) \in \mathbb{B}^m$, the admissible set is

$$X_{\text{adm}}(P) = \{x \in X(P) : \sigma_P(x) = \mathbf{1}_m\}. \quad (12.8)$$

For a SAT instance, the predicates κ_j are clause evaluations; DPLL supplies a classical branching-and-propagation procedure [4], while conflict analysis and learning are central to later solver designs such as GRASP [13]. For a CSP, the predicates encode local relations, and consistency algorithms remove unsupported values or tuples [12]. For graph coloring, they encode edge compatibility; DSATUR is an established problem-specific coloring strategy [1]. In each case, DRCC may choose $\sigma_P(x)$, a restriction of it, or a task-specific quotient as part of $C_P(x)$. However, neither writing the signature nor observing fewer distinct signatures proves that its image can be constructed and searched more cheaply than the corresponding solver.

Routing gives the same distinction. Dynamic programming, including the Held–Karp treatment of sequencing problems, stores recurrence states rather than enumerating complete tours independently [8]; branch-and-bound uses partial routes and bounds. Either method may serve as a baseline or as a reconstruction engine. A DRCC route class has a count-level advantage only after route-class construction, state storage, and reconstruction are charged under the same model as the selected baseline.

12.3 Data Reduction, Symmetry, and Structural Width

Kernelization is an instance-level preprocessing concept. In its standard parameterized form, an input pair (I, k) is transformed into an equivalent pair (I', k') whose encoded size is bounded by a function of the parameter k [7]. By contrast, Equation (12.1) maps candidates or candidate states of a fixed instance into a reduced structural space. A DRCC procedure is therefore not a kernelization theorem unless it also specifies a polynomial-time instance transformation, proves decision equivalence, and proves the required parameter-dependent size bound.

Symmetry breaking removes redundant representatives generated by problem automorphisms or related symmetries; symmetry-breaking predicates provide an established search mechanism for this purpose [2]. DRCC induces the task-dependent relation

$$x \sim_{C_P} y \iff C_P(x) = C_P(y), \quad x, y \in X(P). \quad (12.9)$$

This relation may coincide with symmetry orbits in a particular model, but it need not do so. Its classes are justified by preserved reconstruction information, not by group action alone. Consequently, a DRCC quotient must prove output or admissibility preservation even when no classical symmetry is present.

Width-based methods exploit a decomposition of a graph, hypergraph, or constraint network. Let G_P be a finite interaction graph associated with P , and let $\text{tw}(G_P)$ denote its treewidth. Treewidth and related CSP-width results support tractability statements only when coupled to an explicit decomposition and algorithm [14, 5]. The DRCC quantity $d_{\text{rec}}(P, \zeta)$ is instead the cardinality of one reconstruction fiber. Neither quantity determines the other without a problem-specific theorem. Dynamic programming over a width decomposition may implement reconstruction, but its table construction, state count, and transition cost remain part of the DRCC accounting.

12.4 Tensor Representation and Digital Arithmetic

Let $I(P)$, $J(P)$, and $K(P)$ be finite index sets, and let $\mathcal{V}(P)$ be a finite alphabet. A structural tensor representation in this manuscript is a finite map

$$T_P : I(P) \times J(P) \times K(P) \longrightarrow \mathcal{V}(P). \quad (12.10)$$

For index subsets $I_f \subseteq I(P)$, $J_f \subseteq J(P)$, and $K_f \subseteq K(P)$, the associated tensor restriction is

$$T_f = T_P|_{I_f \times J_f \times K_f}. \quad (12.11)$$

This usage records finite combinatorial relations. It is distinct from canonical, Tucker, or other numerical tensor decompositions and from their algebraic rank notions [10]. Tensor notation alone neither constructs $Z(P)$ nor bounds $N_{\text{Collapse}} + N_{\text{Reconstruction}}$.

Digital arithmetic supplies exact finite examples of value-preserving compression. For the Full Adder developed in Chapter 9, the complete structural chain is

$$\mathbb{B}^3 \xrightarrow{C_{\text{FA}}} \{0, 1, 2, 3\} \xrightarrow{Q_{\text{FA}}} \mathbb{B}^2, \quad G_{\text{FA,struct}} = \frac{8}{4} = 2, \quad (12.12)$$

where C_{FA} is the Hamming-weight collapse and Q_{FA} reconstructs the output pair. The factor 2 is a structural state ratio, not an automatic gate-delay or wall-clock gain.

Wallace and Dadda multipliers are established partial-product reduction architectures [15, 3]. Their compressors preserve the weighted arithmetic value and a final carry-propagate stage reconstructs the product. DRCC uses these architectures as engineering reference cases; it does not claim their invention. A hardware advantage requires a like-for-like implementation study with equal operand width, arithmetic semantics, technology, timing convention, and correctness test, followed by reported delay, area, and power or energy. A reduction in the number of structural rows is not, by itself, any one of those measurements.

12.5 Consolidated Comparison and Evidence Obligations

Table 12.1 records the methodological boundary between established mechanisms and the DRCC interface. ‘‘Possible role’’ means that a method may implement part of a DRCC pipeline; it does not mean that the two methods are equivalent.

Table 12.1: Relation between established methods and DRCC.

Established method	Primary object or action	Possible role in a DRCC procedure	Required distinction or evidence
Exhaustive enumeration	Direct evaluation of candidates in $X(P)$.	Defines a transparent finite baseline and the count $N_A(P)$ when A is enumeration.	A smaller $ Z(P) $ or $ \Omega_P(\zeta) $ is not sufficient; collapse and reconstruction must also be counted.
Backtracking and branch-and-bound	Incremental extension, rejection, and, where applicable, objective bounds.	May construct or search $\Omega_P(\zeta)$; may also be the baseline.	Report visited nodes, predicate checks, bound computations, and ordering rules under the same instance and stopping condition.
Constraint propagation	Domain or relation reduction by local consistency.	May implement collapse, reject impossible reduced states, or support reconstruction.	Local consistency is not identical to global reconstruction; residual search and propagation cost must be reported.
SAT and graph-coloring solvers	Branching, propagation, learning, coloring heuristics, and conflict handling.	Supply domain-specific baselines; their predicates may define a structural signature such as Equation (12.7).	A signature quotient is not a solver. Compare against a named implementation and include construction and search costs.

Continued on next page

Table 12.1 continued

Established method	Primary object or action	Possible role in a DRCC procedure	Required distinction or evidence
Kernelization and data reduction	Polynomial-time instance transformation with equivalence and a parameter-bounded output size.	Motivates preservation-aware preprocessing before reconstruction.	DRCC is a kernelization result only after constructibility, equivalence, parameter, and size-bound obligations are proved.
Symmetry breaking	Removal of equivalent representatives induced by symmetries.	May refine C_P or reduce redundant reconstruction states.	Equality of DRCC states need not be a symmetry orbit; preservation and the cost of detecting symmetry remain separate obligations.
Width methods and dynamic programming	Decomposition into bounded interfaces and reuse of recurrence states.	May provide a reduced state space or an explicit reconstruction algorithm.	Width, table size, d_{rec} , and total runtime are different quantities unless a theorem connects them.
Numerical tensor methods	Multilinear representation, decomposition, approximation, and algebraic rank.	Can supply representation or numerical subroutines when the model requires them.	DRCC structural restrictions and structural classes are not CP/Tucker decompositions or algebraic-rank claims.
Digital arithmetic	Exact gate-level generation, compression, and final arithmetic addition.	Provides finite output-preserving reference systems for collapse and reconstruction.	Hardware claims require equal designs and measured delay, area, power or energy, and correctness; a toy count is not an asymptotic theorem.

For measured runtimes, let $T_A(P) > 0$ be the runtime of baseline A and let $T_{\text{DRCC}}(P, \zeta) > 0$ include both collapse and reconstruction. The two central non-implications are

$$\begin{aligned} |Z(P)| < |X(P)| &\not\Rightarrow N_{\text{DRCC}}(P, \zeta) < N_A(P), \\ N_{\text{DRCC}}(P, \zeta) < N_A(P) &\not\Rightarrow T_{\text{DRCC}}(P, \zeta) < T_A(P). \end{aligned} \tag{12.13}$$

The first separates structural compression from finite operation count; the second separates a unit-cost count from wall-clock behavior. Empirical comparisons must also state the instance set, implementation, hardware, stopping rule, correctness test, repetitions, and aggregation rule; otherwise benchmark design may favor the proposed method unintentionally [9]. An asymptotic claim additionally requires a parameterized family $\{P_n\}_{n \in \mathbb{N}}$, explicit algorithms for every stage, a stated machine or cost model, and a proof of the resulting growth bounds. A finite toy instance cannot provide that proof.

12.6 Novelty Boundary and Conclusion

The literature surveyed above already provides exhaustive search, backtracking, branch-and-bound, consistency enforcement, SAT and coloring solvers, kernelization, symmetry breaking, width-based dynamic programming, numerical tensor decompositions, and Wallace–Dadda multiplier architectures. None is presented as a DRCC invention. The narrower contribution of DRCC-V2 is the common finite interface $X(P)$, C_P , ζ , $\Omega_P(\zeta)$, and a complete collapse-plus-reconstruction account compared with a specified baseline through Proposition 12.1.

Accordingly, this chapter does not claim that DRCC replaces established algorithms, that tensorization proves polynomial solvability, that a structural quotient guarantees runtime gain, that the Full Adder proves an asymptotic result, or that the Dadda multiplier is a DRCC invention. Here P and NP denote the standard deterministic- polynomial and nondeterministic-polynomial decision classes; no claim that $P = NP$ is made. A DRCC comparison is defensible only when preserved information, implementation, complete cost, baseline, and evidence scope are explicit.

13 Scope, Limitations, Open Problems, and Non-Claims

13.1 Scope

This chapter fixes the boundary of the claims supported by the finite DRCC framework. Let P be a finite problem instance, let $X(P)$ be its finite classical candidate space, let $X_{\text{adm}}(P) \subseteq X(P)$ be the admissible candidate set, and let $Z(P)$ be a finite reduced structural space. A specified controlled reduction is a map

$$C_P : X(P) \longrightarrow Z(P). \tag{13.1}$$

For a reduced state $\zeta \in Z(P)$, define the associated reconstruction space and the two finite dimensions by

$$\begin{aligned}\Omega_P(\zeta) &= \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}, \\ d_{\text{frag}}(P) &= |X(P)|, \quad d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|.\end{aligned}\tag{13.2}$$

Let A be a declared baseline algorithm for the same task. Its operation count and measured runtime on P are denoted by $N_A(P) > 0$ and $T_A(P) > 0$, respectively. Let the collapse and reconstruction counts be nonnegative integers, and let their measured times in seconds be nonnegative real numbers. The corresponding DRCC quantities are

$$\begin{aligned}N_{\text{DRCC}}(P, \zeta) &= N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta), \\ T_{\text{DRCC}}(P, \zeta) &= T_{\text{Collapse}}(P) + T_{\text{Reconstruction}}(P, \zeta), \\ G_{\text{count}|A}(P, \zeta) &= \frac{N_A(P)}{N_{\text{DRCC}}(P, \zeta)}, \quad G_{\text{time}|A}(P, \zeta) = \frac{T_A(P)}{T_{\text{DRCC}}(P, \zeta)}.\end{aligned}\tag{13.3}$$

The manuscript therefore distinguishes five claim levels: structural compression, finite count-level gain, measured wall-clock gain, asymptotic improvement for an instance family, and polynomial-time solvability. These levels are not interchangeable. For the last line below, let $(P_n, \zeta_n)_{n \in I}$ be a finite tested family with $I \subseteq \mathbb{N}_{>0}$. In particular,

$$\begin{aligned}d_{\text{rec}}(P, \zeta) < d_{\text{frag}}(P) &\not\Rightarrow G_{\text{count}|A}(P, \zeta) > 1, \\ G_{\text{count}|A}(P, \zeta) > 1 &\not\Rightarrow G_{\text{time}|A}(P, \zeta) > 1,\end{aligned}\tag{13.4}$$

$G_{\text{time}|A}(P_n, \zeta_n) > 1$ on finitely many tested sizes $\not\Rightarrow$ an asymptotic speedup.

Accordingly, Chapters 9–12 establish finite structural identities, model-dependent counts, comparison obligations, and literature relations. They do not establish a universal algorithmic or complexity-theoretic conclusion.

13.2 Formal Limitations

Proposition 13.1 (Structural Compression Is Insufficient for Count Gain). *Let P be a finite instance, let $\zeta \in Z(P)$, and let A be a baseline with $N_A(P) > 0$. The strict structural inequality*

$$0 < d_{\text{rec}}(P, \zeta) < d_{\text{frag}}(P) < \infty\tag{13.5}$$

does not by itself imply $G_{\text{count}|A}(P, \zeta) > 1$.

Proof. Equation (13.5) constrains only the cardinalities of the candidate and reconstruction spaces. It places no upper bound on $N_{\text{Collapse}}(P)$. If, for example, $N_{\text{Collapse}}(P) \geq N_A(P)$, then the non-negativity of the reconstruction count gives

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta) \geq N_A(P).\tag{13.6}$$

Hence $G_{\text{count}|A}(P, \zeta) \leq 1$. Algorithmic cost bounds are therefore additional assumptions, not consequences of structural compression. \square

A second limitation concerns hidden representation cost. Let $\text{enc}_P : Z(P) \rightarrow \{0, 1\}^*$ be a specified encoding of reduced states, where $\{0, 1\}^*$ is the set of finite binary strings. Define the description length

$$\ell_P(\zeta) = |\text{enc}_P(\zeta)|.\tag{13.7}$$

A small value of $|Z(P)|$ does not by itself bound the time needed to construct ζ , the length $\ell_P(\zeta)$, or the consistency work required to reconstruct from it. These costs must be included in N_{Collapse} , $N_{\text{Reconstruction}}$, and their wall-clock counterparts.

Correctness is a separate obligation. Let \mathcal{B} be a benchmark family and \mathcal{O} its declared output space. If

$$A_{\text{base}}, A_{\text{DRCC}} : \mathcal{B} \longrightarrow \mathcal{O}\tag{13.8}$$

are the baseline and DRCC implementations, then every accepted comparison requires

$$A_{\text{DRCC}}(P) = A_{\text{base}}(P) \quad \text{for every tested } P \in \mathcal{B}.\tag{13.9}$$

Here equality means the same decision, admissible output set, or objective value under the benchmark contract. A smaller space is not useful if admissible solutions are lost or the required objective is changed.

Tensor notation has the same limitation. Let $I(P)$, $J(P)$, and $K(P)$ be finite index sets and let \mathcal{A} be a finite alphabet. A structural tensor

$$T_P : I(P) \times J(P) \times K(P) \longrightarrow \mathcal{A} \quad (13.10)$$

is a representation. It is not an algorithm for constructing a useful restriction, selecting a reduced state, or searching $\Omega_P(\zeta)$. Tensorization alone therefore proves neither a count gain nor polynomial-time solvability.

13.3 Empirical and Algorithmic Limitations

The Full-Adder and four-cycle examples are exact finite records. Their value is that all states can be checked, not that their ratios establish large-instance behavior. A small reconstruction space can still be algorithmically difficult to locate, enumerate, verify, or optimize over. All preprocessing, collapse, reconstruction, verification, and postprocessing work must therefore be charged to one of the declared cost components.

A reproducible empirical claim must satisfy the protocol of Chapter 10 and the method-specific comparison rule of Chapter 12. The following ledger states the minimum additional evidence required at this stage.

Table 13.1: Empirical and algorithmic limitations and the evidence required to address them.

Limitation	Required evidence
Finite examples	A parameterized instance family, several non-trivial sizes, and no extrapolation from a single toy instance.
Baseline dependence	A named established baseline for the same task, input, admissibility rule, objective, and output requirement.
Incomplete accounting	Collapse, reconstruction, verification, parsing, preprocessing, and postprocessing assigned to an explicit operation model.
Wall-clock variability	Hardware, software versions, compiler settings, repetitions, aggregation statistic, and timing boundary.
Randomized methods	Random seed policy, instance distribution, stopping rule, and success or approximation criterion.
Resource displacement	Peak memory, stored structural state, and communication or data-movement costs when material.
Digital systems	Identical word width and technology, functional equivalence, and measured delay, area, power, and synthesis conditions relative to Dadda, Wallace, or another declared architecture.

For an indexed family $(P_n)_{n \in I}$, where $I \subseteq \mathbb{N}_{>0}$ is the tested size set, define the measurement record \mathcal{R}_n by

$$\mathcal{R}_n = (d_{\text{frag}}(P_n), d_{\text{rec}}(P_n, \zeta_n), N_{\text{Collapse}}(P_n), N_{\text{Reconstruction}}(P_n, \zeta_n), T_A(P_n), T_{\text{DRCC}}(P_n, \zeta_n)). \quad (13.11)$$

An empirical study must report \mathcal{R}_n for every stated $n \in I$. Persistence of a gain on this finite set is empirical scaling evidence. It is not a worst-case bound, an average-case theorem, or proof of a polynomial-time algorithm.

13.4 Open Problems

The remaining questions can be organized into five research problems. Fix a binary input encoding and let $\ell(P)$ denote the bit length of the encoding of an instance P .

OP1. Characterization of efficiently reconstructible families. For which parameterized families (P_n) do explicit collapse and reconstruction algorithms preserve the required output? More precisely, under a fixed deterministic computation model, does there exist a polynomial $p : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$N_{\text{Collapse}}(P_n) + N_{\text{Reconstruction}}(P_n, \zeta_n) \leq p(\ell(P_n)) \quad \text{for all } n? \quad (13.12)$$

The existence of a small $\Omega_{P_n}(\zeta_n)$ is not sufficient; the algorithms producing and using it must also satisfy the bound.

OP2. Relation to established structural parameters. Let $w(P)$ denote a specified width parameter, such as treewidth or an appropriate CSP width. It remains open whether useful bounds of the form

$$d_{\text{rec}}(P, \zeta) \leq f(w(P), \ell(P)) \quad \text{or} \quad N_{\text{DRCC}}(P, \zeta) \leq g(w(P), \ell(P)) \quad (13.13)$$

hold for non-trivial classes and explicit functions f and g . No equivalence with treewidth, pathwidth, symmetry quotients, kernel size, or CSP width is asserted here [14, 5].

OP3. Lower bounds and failure certificates. Which families force every output-preserving DRCC implementation to pay collapse or reconstruction costs at least as large as a declared baseline? Such lower bounds would identify when structural compression cannot yield an algorithmic gain and would make negative DRCC results as informative as positive ones.

OP4. Transition prediction. Chapter 10 defines a finite transition point $W = (n_w, R_w[s])$ or a tested crossing interval, where n_w is the reported tested size and R_w is the corresponding runtime difference in seconds. It remains open whether n_w can be predicted from measurable structural parameters before complete benchmarking, and whether an observed crossing persists beyond the tested range.

OP5. Reference implementations and benchmark suites. A common suite is still needed for SAT, CSP, graph coloring, routing, selection problems, and digital arithmetic. The digital track should compare functionally equivalent multiplier and adder implementations against established architectures such as Dadda and Wallace [3, 15]; domain-specific evidence must not be promoted to a general complexity claim.

13.5 Non-Claims

The limitations above support one compact boundary statement. This manuscript does not claim:

- a proof of $P = NP$ or $P \neq NP$;
- a universal speedup or superiority over established solvers and hardware architectures;
- polynomial-time solvability from tensor representation, quotienting, or structural compression alone;
- an asymptotic theorem from a finite count table, toy example, or finite benchmark range; or
- that every problem family admits a useful collapse or a small, efficiently searchable reconstruction space.

For completeness, let L^* be a language already known to be NP-complete. The standard conditional implication is

$$(L^* \text{ is NP-complete}) \wedge (L^* \in P \text{ by an explicit correct DRCC algorithm}) \implies P = NP. \quad (13.14)$$

The manuscript supplies no explicit correct polynomial-time DRCC algorithm for any NP-complete language and hence does not supply the second premise of Equation (13.14). Equation (13.14) records a conditional complexity consequence, not a DRCC result.

The final scope of Chapters 9–13 is therefore:

DRCC-V2 provides a finite, method-specific framework for controlled structural reduction, admissible reconstruction, complete cost accounting, and reproducible comparison. Every empirical, asymptotic, or complexity-theoretic extension requires separate algorithms, evidence, and proof.

A DRCC Formula, Symbol, and Verification Register

This appendix is a compact reference for Chapters 9–13. It introduces no additional complexity-theoretic claim. Its purpose is to collect the notation, governing equations, acceptance levels, finite case records, and boundary statements used in the reviewer-oriented revision.

A.1 Core Symbol Register

Let \mathcal{P} be a class of finite problem instances and let $P \in \mathcal{P}$. The symbols used in this appendix are defined in Table A.1.

Table A.1: Core DRCC symbols.

Symbol	Definition and role
$X(P)$	Finite classical candidate space of P .
$X_{\text{adm}}(P)$	Subset of $X(P)$ satisfying the stated admissibility or solution criterion.
$Z(P)$	Finite reduced structural state space.
$C_P : X(P) \rightarrow Z(P)$	Controlled reduction map.
$\zeta \in Z(P)$	Reduced structural state or class.
$\Omega_P(\zeta)$	Admissible reconstruction space associated with ζ .
$d_{\text{frag}}(P)$	Fragmentation dimension, equal to the size of $X(P)$.
$d_{\text{rec}}(P, \zeta)$	Reconstruction dimension, equal to the size of $\Omega_P(\zeta)$.
$R_{\text{struct}}(P)$	Number $ Z(P) $ of reduced structural states.
$N_{\text{Classic}}(P)$	Declared elementary-work count of the selected classical baseline.
$N_{\text{Collapse}}(P)$	Count for constructing or applying the controlled reduction.
$N_{\text{Reconstruction}}(P, \zeta)$	Count for admissible reconstruction and post-reduction evaluation.
$N_{\text{DRCC}}(P, \zeta)$	Total DRCC count, including collapse and reconstruction.
$T_{\text{Classic}}(P)$	Measured wall-clock time of the selected classical baseline.
$T_{\text{DRCC}}(P, \zeta)$	Measured total DRCC wall-clock time.
$G_{\text{count}}(P, \zeta)$	Count-level gain relative to the declared baseline.
$G_{\text{time}}(P, \zeta)$	Empirical wall-clock gain relative to the declared baseline.
$K_{\text{class}}(P)$	Structural class ratio $d_{\text{frag}}(P)/R_{\text{struct}}(P)$.
$K_{\text{fiber}}(P, \zeta)$	Structural fiber ratio $d_{\text{frag}}(P)/d_{\text{rec}}(P, \zeta)$.
$W = (n_w, R_w[\text{s}])$	Exact tested transition point: n_w is the tested size and R_w is the common runtime in seconds at the crossing.

A.2 Core Formula Register

The controlled reduction and its admissible reconstruction fiber are

$$C_P : X(P) \longrightarrow Z(P), \quad \Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \quad (\text{A.1})$$

The two finite dimensions are

$$d_{\text{frag}}(P) = |X(P)|, \quad d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|, \quad 0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \quad (\text{A.2})$$

Whenever the denominators are positive, the structural ratios are

$$K_{\text{class}}(P) = \frac{d_{\text{frag}}(P)}{R_{\text{struct}}(P)}, \quad K_{\text{fiber}}(P, \zeta) = \frac{d_{\text{frag}}(P)}{d_{\text{rec}}(P, \zeta)}. \quad (\text{A.3})$$

These ratios describe structure; they are not runtime gains. Complete count accounting requires

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \quad (\text{A.4})$$

The count-level and empirical time gains are

$$G_{\text{count}}(P, \zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P, \zeta)}, \quad G_{\text{time}}(P, \zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P, \zeta)}. \quad (\text{A.5})$$

A favorable finite count-level regime exists exactly when

$$N_{\text{DRCC}}(P, \zeta) < N_{\text{Classic}}(P) \iff G_{\text{count}}(P, \zeta) > 1. \quad (\text{A.6})$$

A measured wall-clock advantage requires

$$T_{\text{DRCC}}(P, \zeta) < T_{\text{Classic}}(P) \iff G_{\text{time}}(P, \zeta) > 1. \quad (\text{A.7})$$

For a tested family $(P_n)_{n \in I}$, where $I \subseteq \mathbb{N}_{>0}$ is an ordered set of tested sizes, an exact transition point is recorded as

$$G_{\text{time}}(n_w) = 1, \quad W = (n_w, R_w[s]), \quad R_w = T_{\text{Classic}}(P_{n_w}) = T_{\text{DRCC}}(P_{n_w}, \zeta_{n_w}). \quad (\text{A.8})$$

A.3 Evidence and Acceptance Levels

The evidence levels are cumulative. Level **S** is structural, level **C** is complete count-level evidence, and level **E** is reproducible empirical wall-clock evidence.

Level	Minimum evidence required
S	Explicit spaces, map, reconstruction rule, correctness criterion, and relevant finite cardinalities.
C	All requirements of S , plus one declared cost model containing N_{Classic} , N_{Collapse} , and $N_{\text{Reconstruction}}$.
E	All requirements of C , plus reproducible wall-clock measurements under fair hardware, software, parameter, and repetition conditions.

The four DRCC boundary conditions retained for acceptance are:

Rule	Acceptance condition
BC1	The finite candidate space, admissibility criterion, and controlled structural representation are explicit.
BC2	The reduced space $Z(P)$ and the map C_P are constructible by a stated procedure.
BC3	Admissible reconstruction or output preservation is proved or exhaustively verified for the stated instance or family.
BC4	Collapse, reconstruction, state, and measurement costs required by the claimed evidence level are included and remain controlled under the stated model.

A finite structural reduction is accepted under BC1–BC3 and Equation (A.2). A count-level advantage additionally requires BC4 and Equation (A.6). A wall-clock advantage additionally requires fair, reproducible measurements and Equation (A.7). An asymptotic claim requires a parameterized family, explicit algorithms, and proved growth bounds; no finite table substitutes for that proof.

A.4 Case-Study and OPA Register

Table A.2 records the finite result actually supported by each case. A structural ratio is not entered as a runtime gain unless the complete count or time model is available. The mark **S*** denotes a conditional structural record whose missing instance data or construction costs must still be supplied.

Table A.2: Case-study and OPA register.

OPA / case	Finite record	Level	Scope and non-claim
OPA-FA-01	With $\mathbb{B} = \{0, 1\}$, $\mathbb{B}^3 \rightarrow \{0, 1, 2, 3\} \rightarrow \mathbb{B}^2$ and $K_{\text{class}} = 8/4 = 2$.	S	Exact Full-Adder output factorization; output-preserving but not input-invertible. The factor 2 is structural, not an automatic speedup; see Section 9.6.
OPA-DADDA-01	Partial products are compressed through a weighted-value-preserving state and reconstructed by the final adder.	S	Dadda and Wallace multipliers are established architectures, not DRCC inventions. A numerical hardware claim requires equal word width, technology, correctness, delay, area, and power [3, 15]; see Section 9.9.
OPA-C4-01	For the stated fragment of the four-cycle, $d_{\text{frag}} = 3^4 = 81$, $d_{\text{rec}} = 3$, and $K_{\text{fiber}} = 27$.	S	The factor 27 compares a raw candidate space with one reconstruction fiber. Fragment construction and checking costs are not a complete DRCC count; see Section 10.5.1.
OPA-SAT-01	One three-variable formula has 8 assignments, 4 reachable clause-output vectors, and satisfying-fiber size 5.	S	This finite record does not establish a solver advantage or an asymptotic SAT result; see Section 11.4.

Table A.2 continued

OPA / case	Finite record	Level	Scope and non-claim
OPA-CSP-01	A four-variable, three-value record has 81 assignments and an assumed reconstruction size 9.	S*	The value 9 is reproducible only after the local constraint tables are supplied. No count gain is assigned without construction and reconstruction costs.
OPA-TSP-01	A six-city symmetric record has 60 tours, an assumed 18 route classes, and at most 3 representatives per class.	S*	The reconstruction budget 54 is not the total DRCC count. If exactly 54 reconstruction units are used, favorability additionally requires $N_{\text{Collapse}} < 6$.
OPA-HOUSING-01	$N_{\text{Classic}} = 1,208,400$, $N_{\text{DRCC}} = 33,400$, and $G_{\text{count}} \approx 36.18$.	C	Complete finite count-level record under the declared screening and detailed-evaluation model; no wall-clock or universal filtering claim; see Section 11.3.
OPA-ADD16-01	A coupled 16-bit adder has 2^{33} external input triples and at most 2^{17} output sums, so $K_{\text{class}} = 2^{16}$.	S	Internal carries are generated and coupled, not independent external inputs. The structural class ratio is not a runtime gain.

A.5 Boundary and Reference Notes

The literature bridge in Chapter 12 compares DRCC with enumeration, backtracking, branch-and-bound, constraint propagation, SAT and graph coloring solvers, kernelization, symmetry breaking, width methods, numerical tensor methods, and digital arithmetic. These methods remain established baselines or possible components of a DRCC procedure; they are not renamed as DRCC. Complete bibliographic records appear in the manuscript bibliography.

The final boundary statement is:

DRCC-V2 is a finite, method-specific framework for controlled structural reduction and admissible reconstruction. A smaller structural space is not by itself a count theorem; a smaller count is not by itself a wall-clock theorem; and finite evidence is not by itself an asymptotic or complexity-theoretic theorem. DRCC-V2 does not claim a universal speedup, polynomial solvability from tensorization alone, or a proof of $P = NP$ or $P \neq NP$.

Accordingly, the manuscript supports only conclusions whose evidence level, baseline, domain, cost model, and tested range are stated. Any stronger empirical, asymptotic, or complexity-theoretic conclusion requires a separate algorithm, experiment, or proof.

Bibliography

- [1] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.
- [2] James M. Crawford, Matthew L. Ginsberg, Eugene M. Luks, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 148–159. Morgan Kaufmann, 1996.
- [3] Luigi Dadda. Some schemes for parallel multipliers. *Alta Frequenza*, 34:349–356, 1965.
- [4] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [5] Eugene C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 4–9. AAAI Press, 1990.
- [6] Solomon W. Golomb and Leonard D. Baumert. Backtrack programming. *Journal of the ACM*, 12(4):516–524, 1965.
- [7] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007.
- [8] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.
- [9] John N. Hooker. Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1(1):33–42, 1995.
- [10] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [11] Ailsa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.
- [12] Alan K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.
- [13] João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, 1999.
- [14] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.
- [15] C. S. Wallace. A suggestion for a fast multiplier. *IEEE Transactions on Electronic Computers*, EC-13(1):14–17, 1964.