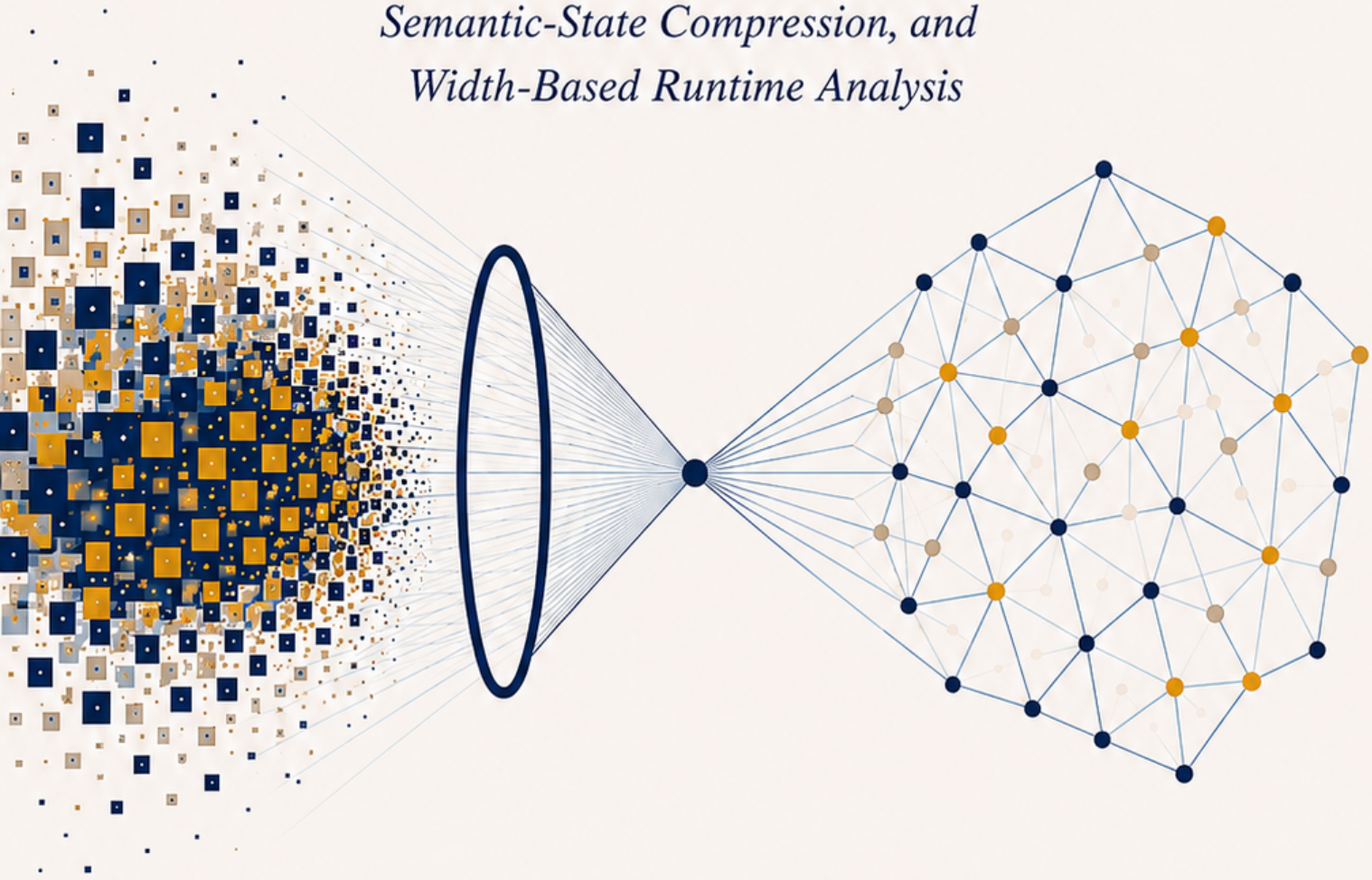
CPR-WIDTH

Controlled Partial Reconstruction Width

A Structural Framework for
Active Reconstruction Interfaces,
Semantic-State Compression, and
Width-Based Runtime Analysis



$$\omega_{\mathrm{rec}}(P,\pi) \longrightarrow S_{\mathrm{CPR}}(P,\pi) \longrightarrow N_{\mathrm{CPR}}(P,\pi).$$

REZA HESAMIY

MUNICH, GERMANY

2026

CPR-WIDTH-V2.0

CPR-WIDTH

Controlled Partial Reconstruction Width

A Structural Framework for Active Reconstruction Interfaces,
Semantic-State Compression, and
Width-Based Runtime Analysis

The Theory of Controlled and Constant CPR-Width

Reza Hesamiy

CPR-WIDTH-V2.0

Munich, 2026

Abstract

Controlled Partial Reconstruction Width (*CPR-Width*) is a reconstruction-oriented structural parameter for finite combinatorial problems. Instead of measuring the complete candidate space, CPR-Width measures the maximum number of task-relevant components that must remain simultaneously active during a sound and complete reconstruction process.

Let P be a finite problem instance and let π be an admissible reconstruction ordering. At each reconstruction stage, an active interface records the processed components whose information can still influence an admissible future continuation. CPR-Width is defined from the maximum cardinality of this interface over the complete reconstruction process and is minimized over admissible orderings.

The framework distinguishes raw interface states, reachable states, and task-relative semantic states obtained through future-extension equivalence, yielding the structural chain $\omega_{\mathrm{rec}}(P,\pi) \to S_{\mathrm{CPR}}(P,\pi) \to N_{\mathrm{CPR}}(P,\pi)$ connecting reconstruction width, semantic-state growth, and complete computational cost. A full runtime model separately accounts for construction, reduction, semantic transition, reconstruction, verification, and output costs. Four boundary conditions are then stated for explicit representation, polynomial constructibility, sound and complete reconstruction, and polynomial semantic-state control.

Representative realizations are given for Boolean satisfiability, constraint satisfaction, graph coloring, tensor representations, and digital arithmetic, including a dedicated theory of controlled and constant CPR-Width subclasses. Exact finite verification and a family-level proof are given for the full-adder and ripple-adder models.

This work establishes a formal framework for analyzing controlled partial reconstruction. It does not establish universal tractability, a general polynomial-time algorithm for NP-complete problems, or a proof of $P = NP$.

Keywords: Controlled Partial Reconstruction Width; CPR-Width; active reconstruction interface; semantic-state compression; future-extension equivalence; structural width; finite combinatorial reconstruction; runtime analysis; constraint satisfaction; graph coloring.

Scope and Non-Claims. The results apply only to the explicitly defined reconstruction models and to problem families satisfying all stated boundary conditions and runtime assumptions. A small reconstruction width alone does not imply polynomial runtime. Structural compression alone does not imply a wall-clock advantage. No NP-complete language is proved in this work to satisfy the complete CPR tractability conditions.

Contents

1 Introduction and Motivation

Classical combinatorial computation is traditionally described in terms of complete candidate spaces. A computational problem is represented by the set of all admissible assignments, configurations, routes, colorings, certificates, or other candidate objects, and algorithmic complexity is usually analyzed with respect to the size of this global search space. From the viewpoint of reconstruction, however, the complete candidate space is often not the quantity that determines the essential computational difficulty.

During a reconstruction process, previously processed information gradually loses its relevance. At every reconstruction stage, some information may safely be discarded because it can no longer influence any admissible continuation, whereas other information must remain available until reconstruction has been completed. The mathematical distinction between relevant and irrelevant information is therefore fundamental.

The present manuscript studies the amount of information that must remain simultaneously available during a sound and complete reconstruction. Rather than measuring the size of the complete search space, the theory developed here measures the size of the active reconstruction interface. This quantity is called the *Controlled Partial Reconstruction Width (CPR-WIDTH)*.

Unlike many classical structural parameters, CPR-WIDTH is not defined directly on graphs or decomposition trees. Classical graph-width parameters such as treewidth and pathwidth relate structural restrictions of graphs to algorithmic tractability [14, 2]. Parameterized complexity extends this perspective by separating the influence of the input size from the influence of additional structural parameters [8, 4]. CPR-WIDTH is introduced in relation to this line of work, but it is not defined as a graph-decomposition parameter. Rather, it is a reconstruction-oriented width notion whose value depends on the active information maintained during a declared reconstruction process. A formal structural comparison between CPR-Width and classical width parameters is posed as an open problem in Chapter 9, together with the precise conditions any such comparison theorem would have to fix.

Its value depends on the chosen reconstruction ordering, the active reconstruction interface, the declared computational task, and the semantic information required for every admissible future continuation. Consequently, two reconstruction procedures operating on the same underlying combinatorial structure may possess different reconstruction widths whenever their active interfaces differ.

The objective of this manuscript is therefore not to replace existing complexity theory, but to introduce a reconstruction-oriented structural parameter that permits a mathematical analysis of active information, semantic-state growth, and reconstruction complexity. The theory developed in the following chapters proceeds through three successive levels: first, the active reconstruction interface is introduced as the central structural object; second, the reachable semantic-state space generated by this interface is studied; finally, the resulting computational effort required by controlled reconstruction is analyzed.

The central mathematical relationship developed throughout this manuscript is

$$\boldsymbol{\omega_{\mathrm{rec}}(P,\pi)} \longrightarrow \boldsymbol{S_{\mathrm{CPR}}(P,\pi)} \longrightarrow \boldsymbol{N_{\mathrm{CPR}}(P,\pi)} \tag{1.1}$$

The first quantity measures the maximal active reconstruction interface, the second describes the corresponding semantic-state space, and the third represents the computational effort required by the declared reconstruction model. The formal construction of these quantities is developed progressively in Chapters 2–4.

No statement in this manuscript should be interpreted as a proof of $P = NP$, as a universal polynomial-time algorithm for NP-complete problems, or as evidence that structural reduction alone implies tractability. Instead, the objective is to develop a mathematically rigorous theory of Controlled Partial Reconstruction Width together with its structural, semantic, and computational consequences.

The remaining chapters develop this theory systematically. Chapter 2 introduces the reconstruction interface and the formal definition of CPR-WIDTH. Chapter 3 develops the corresponding semantic-state framework. Chapter 4 establishes the complete runtime model. Chapter 5 derives the boundary conditions required for tractability. Chapters 6–8 illustrate the theory by representative problem classes, a dedicated theory of controlled and constant CPR-Width subclasses, and finite verification records. Chapter 9 discusses the resulting complexity-theoretic scope, open problems, and future research directions.

2 Controlled Partial Reconstruction Width

This chapter introduces the central structural parameter of the CPR-WIDTH framework: how many processed components must remain simultaneously available during a sound, complete, and task-relative reconstruction process. The construction proceeds in the following non-circular order:

$$\mathcal{R}_i^{\pi} \longrightarrow E_i^{\pi}(r) \longrightarrow \mathrm{Sig}_{\tau,i}^{\pi}(r) \longrightarrow B_i^{\pi,\tau} \longrightarrow \omega_{\mathrm{rec}}(P,\pi,\tau).$$

Thus, reachable prefix states and their future behavior are defined before the active reconstruction interface is introduced.

2.1 Declared CPR Reconstruction Model

Let P be a finite problem instance equipped with a finite reconstruction-component representation $V(P) = \{v_1, \ldots, v_n\}$. The component representation is part of the declared reconstruction model; it is not assumed to be a canonical representation of the abstract problem instance. Every component $v \in V(P)$ has a finite nonempty domain D_v, $\varnothing \neq D_v$, $|D_v| < \infty$. The complete assignment space is $\mathcal{X}(P) = \prod_{v \in V(P)} D_v$, and $\mathrm{Sol}(P) \subseteq \mathcal{X}(P)$ denotes the set of complete feasible reconstructions. A declared computational task is denoted by τ; typical tasks include decision, counting, enumeration, optimization, and exact output reconstruction.

Definition 2.1 (Declared CPR Reconstruction Model)**.** A declared CPR reconstruction model for P and task τ is a tuple

$$\mathfrak{M}(P,\tau) = (\mathrm{Adm}_{P,\tau}, \mathcal{R}, \Lambda, \delta, \mathrm{Sig}_{\tau}),$$

where $\mathrm{Adm}_{P,\tau}$ is the admissibility predicate for reconstruction orderings, $\mathcal{R}$ specifies the reachable prefix-state spaces, Λ specifies the admissible next reconstruction values, δ specifies the stage-to-stage transition maps, and Sig_{τ} specifies a reconstruction-safe, task-relative future-signature family.

All widths in this chapter are relative to the fixed problem representation, the declared task, and the declared reconstruction model.

2.2 Admissible Reconstruction Orderings

A reconstruction ordering is a permutation $\pi = (v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)})$.

Definition 2.2 (Admissible Reconstruction Ordering)**.** An ordering π is admissible for the model $\mathfrak{M}(P,\tau)$ if $\mathrm{Adm}_{P,\tau}(\pi) = 1$. The admissibility predicate must encode all declared structural, transition, representation, and task-preservation conditions required by the reconstruction model.

The set of admissible reconstruction orderings is

$$\Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M}) = \{\pi \in \mathrm{Sym}(V(P)) : \mathrm{Adm}_{P,\tau}(\pi) = 1\}.$$

Assume throughout that $\Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M}) \neq \varnothing$. When the task and reconstruction model are fixed, the shorter notation $\Pi_{\mathrm{adm}}(P)$ may be used.

2.3 Processed Prefix and Unresolved Suffix

Let $\pi \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})$ be fixed. After reconstruction stage i, the processed prefix is $V_i^{\pi} = \{v_{\pi(1)}, \ldots, v_{\pi(i)}\}$, $0 \le i \le n$, and the unresolved suffix is $\overline{V_i^{\pi}} = V(P) \setminus V_i^{\pi}$. The boundary cases are $V_0^{\pi} = \varnothing$, $V_n^{\pi} = V(P)$, and correspondingly $\overline{V_0^{\pi}} = V(P)$, $\overline{V_n^{\pi}} = \varnothing$. The complete prefix-assignment space at stage i is $\mathcal{A}_i^{\pi}(P) = \prod_{v \in V_i^{\pi}} D_v$, with the empty Cartesian product interpreted as the singleton containing the empty assignment, $\mathcal{A}_0^{\pi}(P) = \{\varnothing\}$. The complete suffix-assignment space at stage i is $\mathcal{C}_i^{\pi}(P) = \prod_{v \in \overline{V_i^{\pi}}} D_v$, so $\mathcal{C}_n^{\pi}(P) = \{\varnothing\}$.

2.4 Reachable Prefix States

Definition 2.3 (Reachable Prefix-State Space)**.** For every reconstruction stage i, the reachable prefix-state space is a set $\mathcal{R}_i^\pi(P) \subseteq \mathcal{A}_i^\pi(P)$. A prefix assignment belongs to $\mathcal{R}_i^\pi(P)$ if it can be generated after the first i reconstruction steps under the declared transition and admissibility rules of $\mathfrak{M}(P,\tau)$.

Initially, $\mathcal{R}_0^\pi(P) = \{\varnothing\}$. Reachability does not imply extendability: a reachable prefix state may satisfy all local conditions tested so far while still admitting no feasible complete continuation. For $r \in \mathcal{R}_i^\pi(P)$ and $B \subseteq V_i^\pi$, the restriction of r to the components in B is denoted by $r|_B$.

2.5 Future Continuations and Task Evaluation

For a reachable prefix state $r \in \mathcal{R}_i^\pi(P)$, define its feasible continuation set by

$$E_i^\pi(r) = \{s \in \mathcal{C}_i^\pi(P) : r \cup s \in \mathrm{Sol}(P)\}.$$

At a fixed stage i, all sets $E_i^\pi(r)$ are subsets of the same continuation universe $\mathcal{C}_i^\pi(P)$. The declared task τ determines a stage-relative evaluation map $\Theta_{\tau,i}^\pi : \mathcal{R}_i^\pi(P) \times \mathcal{P}(\mathcal{C}_i^\pi(P)) \to Y_{\tau,i}$, and the task-relative future value of a reachable state is $\mathrm{Ans}_{\tau,i}^\pi(r) = \Theta_{\tau,i}^\pi(r, E_i^\pi(r))$. For a decision task, one may use $\mathrm{Ans}_{\mathrm{dec},i}^\pi(r) = \mathbf{1}[E_i^\pi(r) \neq \varnothing]$; for counting, $\mathrm{Ans}_{\mathrm{count},i}^\pi(r) = |E_i^\pi(r)|$; for enumeration, $\mathrm{Ans}_{\mathrm{enum},i}^\pi(r) = \{r \cup s : s \in E_i^\pi(r)\}$. For an optimization task, the map must specify whether the required object is an optimum value, an optimal witness, the set of all optimal witnesses, or another declared optimization certificate.

2.6 Stage Transitions

For every nonterminal stage $i < n$, define the next component by $w_i^\pi = v_{\pi(i+1)}$. For a reachable prefix state $r \in \mathcal{R}_i^\pi(P)$, the set of admissible next values is

$$\Lambda_i^\pi(r) = \left\{a \in D_{w_i^\pi} : r \cup \{w_i^\pi \mapsto a\} \in \mathcal{R}_{i+1}^\pi(P)\right\},$$

and for every $a \in \Lambda_i^\pi(r)$ the transition map is $\delta_i^\pi(r,a) = r \cup \{w_i^\pi \mapsto a\} \in \mathcal{R}_{i+1}^\pi(P)$.

2.7 Reconstruction-Safe Future Signatures

A task value alone may be too coarse for a stage-wise reconstruction process: two decision states may both admit a feasible continuation while requiring different admissible next values. The active interface must therefore preserve not only the current task answer but also the transition behavior required for future reconstruction.

Definition 2.4 (Reconstruction-Safe Future-Signature Family)**.** A family of maps $\mathrm{Sig}_{\tau,i}^\pi : \mathcal{R}_i^\pi(P) \to \Sigma_{\tau,i}^\pi$, $0 \le i \le n$, is called reconstruction-safe if the following conditions hold.

S1 – Task preservation. For all $r, r' \in \mathcal{R}_i^\pi(P)$, $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$ implies $\mathrm{Ans}_{\tau,i}^\pi(r) = \mathrm{Ans}_{\tau,i}^\pi(r')$.

S2 – Preservation of admissible next values. For every $i < n$, $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$ implies $\Lambda_i^\pi(r) = \Lambda_i^\pi(r')$.

S3 – Successor congruence. If $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$ and $a \in \Lambda_i^\pi(r) = \Lambda_i^\pi(r')$, then $\mathrm{Sig}_{\tau,i+1}^\pi(\delta_i^\pi(r,a)) = \mathrm{Sig}_{\tau,i+1}^\pi(\delta_i^\pi(r',a))$.

Condition S1 preserves the declared task-relative future value. Conditions S2 and S3 guarantee that replacing a prefix state by another state with the same signature does not make the next reconstruction step ambiguous. The identity signature $\mathrm{Sig}_{\tau,i}^\pi(r) = r$ is always reconstruction-safe, so a reconstruction-safe future-signature family always exists, although it may provide no compression. A constant signature is admissible only when it satisfies S1–S3; consequently, the formal model cannot collapse every state to one signature unless task values and transition behavior are genuinely indistinguishable.

2.8 Task-Sufficient Active Reconstruction Interface

Definition 2.5 (Task-Sufficient Interface)**.** Let i be a reconstruction stage. A set $B \subseteq V_i^\pi$ is called a task-sufficient reconstruction interface if

$$\forall r, r' \in \mathcal{R}_i^\pi(P): \qquad r|_B = r'|_B \Longrightarrow \mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r').$$

This condition guarantees that the complete reconstruction-safe future signature is determined by the values retained on B; all processed components outside B may be forgotten jointly without changing the declared task-relative and transition-relative future behavior. The family of all task-sufficient interfaces at stage i is denoted $\mathfrak{B}_i^{\pi,\tau}(P)$.

Lemma 2.1 (Existence of a Task-Sufficient Interface)**.** *For every reconstruction stage i, $\mathfrak{B}_i^{\pi,\tau}(P) \neq \varnothing$.*

Proof. The complete processed prefix V_i^π is task-sufficient: if $r|_{V_i^\pi} = r'|_{V_i^\pi}$, then $r = r'$, so $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$. Therefore $V_i^\pi \in \mathfrak{B}_i^{\pi,\tau}(P)$. □

2.9 Interface Factorization

For $B \subseteq V_i^\pi$, define the coordinate projection $\rho_{i,B}^\pi : \mathcal{R}_i^\pi(P) \to \prod_{v\in B} D_v$ by $\rho_{i,B}^\pi(r) = r|_B$.

Proposition 2.1 (Interface Factorization)**.** *A set $B \subseteq V_i^\pi$ is task-sufficient if and only if there exists a unique map $\overline{\mathrm{Sig}}_{\tau,i,B}^\pi : \rho_{i,B}^\pi(\mathcal{R}_i^\pi(P)) \to \Sigma_{\tau,i}^\pi$ such that $\mathrm{Sig}_{\tau,i}^\pi = \overline{\mathrm{Sig}}_{\tau,i,B}^\pi \circ \rho_{i,B}^\pi$.*

Proof. Assume first that B is task-sufficient. For $z \in \rho_{i,B}^\pi(\mathcal{R}_i^\pi(P))$, choose a reachable state r satisfying $\rho_{i,B}^\pi(r) = z$ and define $\overline{\mathrm{Sig}}_{\tau,i,B}^\pi(z) = \mathrm{Sig}_{\tau,i}^\pi(r)$. If r' is another reachable state satisfying $\rho_{i,B}^\pi(r') = z$, then $r|_B = r'|_B$, and task sufficiency gives $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$; hence the map is well defined. Conversely, assume the factorization exists. If $r|_B = r'|_B$, then $\rho_{i,B}^\pi(r) = \rho_{i,B}^\pi(r')$, so $\mathrm{Sig}_{\tau,i}^\pi(r) = \overline{\mathrm{Sig}}_{\tau,i,B}^\pi(\rho_{i,B}^\pi(r)) = \overline{\mathrm{Sig}}_{\tau,i,B}^\pi(\rho_{i,B}^\pi(r')) = \mathrm{Sig}_{\tau,i}^\pi(r')$, so B is task-sufficient. Uniqueness follows because every element of the domain is the projection of at least one reachable prefix state. □

2.10 Minimum Active Interface

A task-sufficient interface need not be unique: different component subsets may preserve exactly the same reconstruction-safe signature. The central width parameter is therefore defined through the minimum required cardinality rather than through a supposedly unique interface.

Definition 2.6 (Minimum Task-Sufficient Interface Size)**.** The minimum task-sufficient interface size at stage i is

$$b_i^{\pi,\tau}(P) = \min\left\{|B| : B \in \mathfrak{B}_i^{\pi,\tau}(P)\right\}.$$

The minimum exists because $\mathfrak{B}_i^{\pi,\tau}(P) \neq \varnothing$ and V_i^π is finite. An active reconstruction interface may be chosen as any cardinality-minimal task-sufficient set, $B_i^{\pi,\tau} \in \arg\min_{B\in\mathfrak{B}_i^{\pi,\tau}(P)} |B|$. The minimizing set need not be unique, but every minimizing set has the same cardinality $|B_i^{\pi,\tau}| = b_i^{\pi,\tau}(P)$; accordingly, the phrase "components that must remain active" refers to the minimum cardinality $b_i^{\pi,\tau}(P)$, not to one uniquely determined component set. The complete selected interface sequence is $\mathcal{B}(P,\pi,\tau) = (B_0^{\pi,\tau}, B_1^{\pi,\tau}, \ldots, B_n^{\pi,\tau})$. Since $\mathcal{R}_0^\pi(P) = \{\varnothing\}$, one has $b_0^{\pi,\tau}(P) = 0$ and the initial interface may be selected as $B_0^{\pi,\tau} = \varnothing$. A terminal empty interface is not imposed automatically: it is valid only when all required output information has already been materialized or when the terminal signature is independent of the processed component values.

2.11 Why the Componentwise Test Is Insufficient

The following observation explains why a former componentwise definition cannot serve as the general definition of an active interface.

Lemma 2.2 (Single-Component Necessity Witness)**.** *Let $v \in V_i^\pi$. Assume that there exist states $r, r' \in \mathcal{R}_i^\pi(P)$ such that $r|_{V_i^\pi \setminus \{v\}} = r'|_{V_i^\pi \setminus \{v\}}$, but $\mathrm{Sig}_{\tau,i}^\pi(r) \neq \mathrm{Sig}_{\tau,i}^\pi(r')$. Then every task-sufficient interface contains v.*

Proof. Let B be task-sufficient and suppose $v \notin B$. Then $B \subseteq V_i^\pi \setminus \{v\}$, so $r|_B = r'|_B$. Task sufficiency would imply $\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r')$, contradicting the hypothesis. Therefore $v \in B$. □

Remark 2.1 (Failure of the Converse)**.** The converse of the preceding lemma is false. Suppose that two processed binary components a, b have only the reachable prefix states $\mathcal{R}_i^\pi(P) = \{00, 11\}$, with $\mathrm{Sig}_{\tau,i}^\pi(00) \neq \mathrm{Sig}_{\tau,i}^\pi(11)$. There are no two reachable states that differ only in a, and none that differ only in b. Nevertheless, the empty interface is not task-sufficient, because it would identify the two states 00 and 11; at least one of the components a or b must be retained. Therefore, a componentwise difference test is a valid necessity witness, but it is not a complete interface definition.

2.12 Controlled Partial Reconstruction Width

The unnormalized interface size is the primary operational quantity.

Definition 2.7 (Unnormalized CPR Interface Width)**.** For a fixed problem P, admissible ordering π, task τ, and reconstruction model $\mathfrak{M}$, define

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = \max_{0 \le i \le n} b_i^{\pi,\tau}(P).$$

The value $\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$ is the actual minimum number of components that must remain simultaneously active at the widest reconstruction stage. This unnormalized quantity is used directly in every bound proved in Chapters 3–4.

Definition 2.8 (Controlled Partial Reconstruction Width)**.** The normalized Controlled Partial Reconstruction Width is

$$\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = \max\left\{0,\, \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) - 1\right\}.$$

Both forms are used consistently throughout this manuscript, for two distinct purposes. The unnormalized width $\widehat{\omega}_{\mathrm{rec}}$ is the operational quantity: it equals the actual interface cardinality and is the form used in every state-count and runtime bound of Chapters 3 and 4. The normalized width $\omega_{\mathrm{rec}} = \max\{0, \widehat{\omega}_{\mathrm{rec}} - 1\}$ is the quantity used whenever CPR-Width is compared, as a width parameter in the usual sense, with interface-based width notions whose convention likewise sets the width of a single-component interface to zero rather than one. Every occurrence of ω_{rec} in this manuscript, in particular in the boundary conditions of Chapter 5 and the tractability statements of Chapter 9, refers to this normalized quantity; every underlying cardinality bound is proved for $\widehat{\omega}_{\mathrm{rec}}$ first and then restated for ω_{rec} using the identity $\widehat{\omega}_{\mathrm{rec}} = \omega_{\mathrm{rec}} + 1$ for $\widehat{\omega}_{\mathrm{rec}} \geq 1$. Accordingly, $\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = 0$ may still mean that one component must remain active. When τ and $\mathfrak{M}$ are fixed, the shorter forms $\widehat{\omega}_{\mathrm{rec}}(P, \pi)$ and $\omega_{\mathrm{rec}}(P, \pi)$ may be used.

2.13 Interface Bound

Lemma 2.3 (Interface Bound)**.** *For every stage i, $b_i^{\pi,\tau}(P) \leq \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$. Moreover, $\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \leq \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1$. Consequently, for every selected minimum active interface, $|B_i^{\pi,\tau}| \leq \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1$.*

Proof. The first inequality follows immediately from $\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = \max_{0 \le j \le n} b_j^{\pi,\tau}(P)$. Let $M = \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$. If $M = 0$, then $M = 0 \leq 1 = \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1$. If $M \geq 1$, then $\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = M - 1$, and therefore $M = \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1$. Finally, $|B_i^{\pi,\tau}| = b_i^{\pi,\tau}(P)$, because $B_i^{\pi,\tau}$ is cardinality-minimal. □

2.14 Minimum CPR-Width over Admissible Orderings

Because $\Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M}) \neq \varnothing$, the following minimum is well defined.

Definition 2.9 (Minimum Controlled Partial Reconstruction Width)**.** The minimum Controlled Partial Reconstruction Width of P, relative to task τ and model $\mathfrak{M}$, is

$$\omega_{\mathrm{rec}}(P, \tau; \mathfrak{M}) = \min_{\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})} \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}).$$

The corresponding unnormalized minimum is $\widehat{\omega}_{\mathrm{rec}}(P, \tau; \mathfrak{M}) = \min_{\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})} \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$. When the task and reconstruction model are fixed, these quantities may be written as $\omega_{\mathrm{rec}}(P)$ and $\widehat{\omega}_{\mathrm{rec}}(P)$. The minimization over orderings does not silently alter the component representation $V(P)$, the local domains D_v, the declared task τ, the admissibility predicate, the reachable-state rules, the transition system, or the reconstruction-safe future-signature family: only the admissible reconstruction ordering is varied.

2.15 Ordering Dependence

Proposition 2.2 (Ordering Dependence)**.** *There exist a finite problem instance P, a fixed task τ, a fixed reconstruction model $\mathfrak{M}$, and two admissible orderings π_1, π_2 such that $\omega_{\mathrm{rec}}(P, \pi_1, \tau; \mathfrak{M}) \neq \omega_{\mathrm{rec}}(P, \pi_2, \tau; \mathfrak{M})$.*

Proof. Let $V(P) = \{a, b, c\}$ with binary domains $D_a = D_b = D_c = \{0, 1\}$. Impose the constraints $a = c$ and $b = c$, so that $\mathrm{Sol}(P) = \{000, 111\}$. Let the declared task be decision. A partial assignment is reachable if every constraint whose complete scope has already been processed is satisfied. The reconstruction-safe signature records whether a feasible complete continuation exists, the set of admissible next values, and the signatures of the corresponding successor states.

Consider first the ordering $\pi_1 = (a, b, c)$. After stage 2, neither equality constraint has been completely processed, because both contain c. Therefore $\mathcal{R}_2^{\pi_1}(P) = \{00, 01, 10, 11\}$. The admissible next-value sets for c are $\Lambda_2^{\pi_1}(00) = \{0\}$, $\Lambda_2^{\pi_1}(11) = \{1\}$, and $\Lambda_2^{\pi_1}(01) = \Lambda_2^{\pi_1}(10) = \varnothing$. The singleton interface $\{a\}$ is not sufficient, because the states 00 and 01 agree on a but have different admissible next-value sets; the singleton interface $\{b\}$ is not sufficient, because the states 00 and 10 agree on b but have different admissible next-value sets. Hence $b_2^{\pi_1, \mathrm{dec}}(P) = 2$. At every other stage, an interface of size at most one is sufficient. Therefore $\widehat{\omega}_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) = 2$ and $\omega_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) = 1$.

Now consider the ordering $\pi_2 = (a, c, b)$. After stage 2, the constraint $a = c$ has already been processed, so the inconsistent prefixes are removed and $\mathcal{R}_2^{\pi_2}(P) = \{00, 11\}$. The admissible next-value sets for b are $\Lambda_2^{\pi_2}(00) = \{0\}$ and $\Lambda_2^{\pi_2}(11) = \{1\}$. Either component a or component c distinguishes these two reachable states, so $b_2^{\pi_2, \mathrm{dec}}(P) = 1$. At every stage, an interface of size at most one is task-sufficient. Hence $\widehat{\omega}_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M}) = 1$ and $\omega_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M}) = 0$.

Therefore $\omega_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) \neq \omega_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M})$. □

2.16 Structural Interpretation

Controlled Partial Reconstruction Width does not measure the total size of the complete candidate space. The unnormalized quantity $\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$ measures the minimum maximal number of processed components whose values must remain simultaneously available under the fixed representation, task, transition rules, and reconstruction-safe signature family; the normalized quantity $\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$ is obtained from it by the minus-one convention explained above. A task-sufficient interface satisfies $r|_B = r'|_B \Rightarrow \mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r')$; equivalently, the future-signature map factors through the interface projection, $\mathrm{Sig}_{\tau,i}^{\pi} = \overline{\mathrm{Sig}}_{\tau,i,B}^{\pi} \circ \rho_{i,B}^{\pi}$. This factorization is the formal statement that all jointly forgotten components are semantically and transitionally unnecessary for the declared reconstruction task.

CPR-Width therefore depends on the component representation, the component domains, the re-

construction ordering, the declared computational task, the reachability and transition rules, and the reconstruction-safe future-signature family. The same finite problem instance may therefore possess different CPR-Width values for decision, counting, enumeration, optimization, or exact reconstruction tasks; in particular, $B_i^{\pi,\mathrm{dec}} \neq B_i^{\pi,\mathrm{count}}$ and $\omega_{\mathrm{rec}}(P, \pi, \mathrm{dec}; \mathfrak{M}) \neq \omega_{\mathrm{rec}}(P, \pi, \mathrm{count}; \mathfrak{M})$ may occur. CPR-Width counts active components; it does not directly measure their bit-level information content. Local domain sizes, projected interface-state spaces, reachable interface states, and semantic quotient states are developed in the following chapter, together with the architecture overview of Figure 2.1.

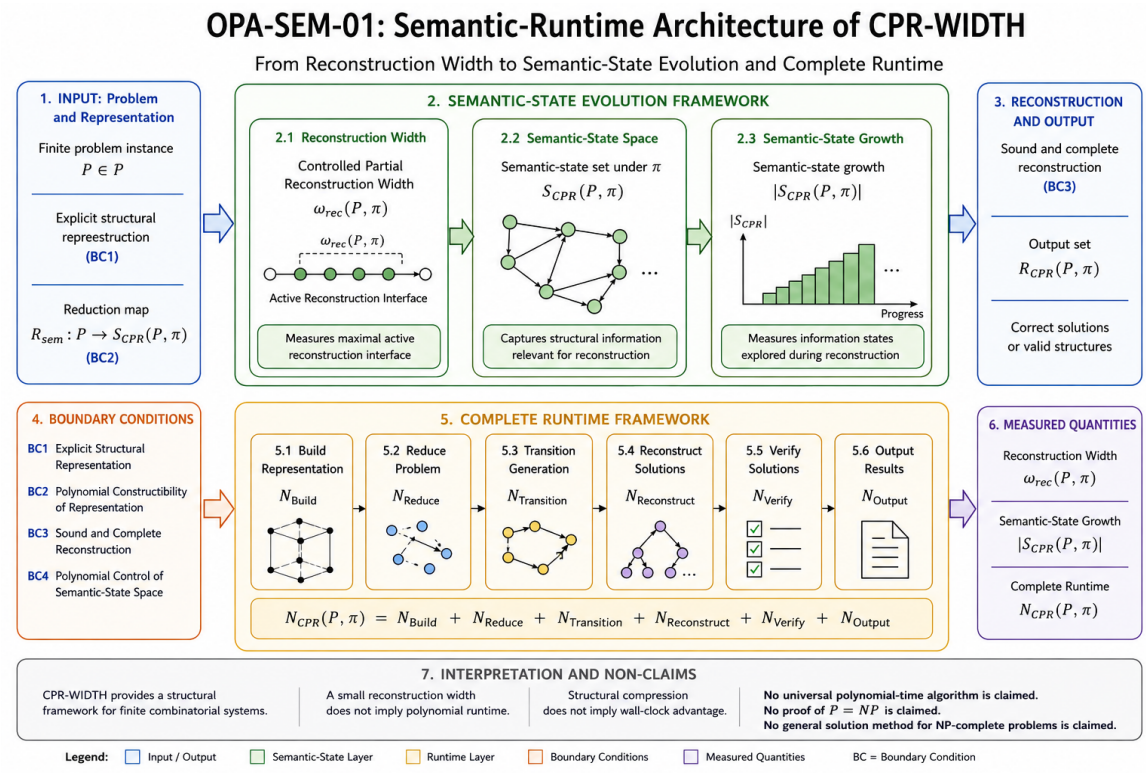


Figure 2.1: OPA-SEM-01: Semantic-runtime architecture of CPR-WIDTH. The figure summarizes the transition from reconstruction width $\omega_{\text{rec}}(P, \pi)$ to semantic-state growth $S_{\text{CPR}}(P, \pi)$ and the complete operation-count model $N_{\text{CPR}}(P, \pi)$.

3 Raw States, Semantic States, and Reconstruction

Chapter 2 introduced reachable prefix states, task-sufficient active interfaces, reconstruction-safe future signatures, and Controlled Partial Reconstruction Width. The present chapter studies the state spaces induced by these objects, proceeding through four distinct levels,

$$\mathcal{R}_i^{\pi}(P) \longrightarrow U_i(P,\pi,\tau;\mathfrak{M}) \longrightarrow U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \longrightarrow S_i(P,\pi,\tau;\mathfrak{M}),$$

where $\mathcal{R}_i^{\pi}(P)$ is the complete reachable prefix-state space, $U_i(P,\pi,\tau;\mathfrak{M})$ is the raw assignment space of the active interface, $U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})$ is the projection of reachable prefix states onto that interface, and $S_i(P,\pi,\tau;\mathfrak{M})$ is the semantic quotient induced by the reconstruction-safe signature. Throughout, the problem instance P, the declared task τ, the reconstruction model $\mathfrak{M}$, an admissible ordering $\pi \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})$, and one cardinality-minimal task-sufficient interface $B_i^{\pi,\tau}$ at every stage are fixed; dependence on τ and $\mathfrak{M}$ may be suppressed.

3.1 Raw Interface States

At stage i, the active interface is $B_i^{\pi,\tau} \subseteq V_i^{\pi}$; every active component $v \in B_i^{\pi,\tau}$ has the finite nonempty domain D_v introduced in Chapter 2.

Definition 3.1 (Raw Interface-State Space)**.** The raw interface-state space at stage i is $U_i(P,\pi,\tau;\mathfrak{M}) = \prod_{v\in B_i^{\pi,\tau}} D_v$.

When the task and model are fixed, the shorter notation $U_i(P,\pi)$ may be used. The space $U_i(P,\pi,\tau;\mathfrak{M})$ contains every syntactically possible assignment to the selected active interface; it does not assert that every such assignment can actually occur during the declared reconstruction process. If $B_i^{\pi,\tau} = \varnothing$, the empty Cartesian product is interpreted as a singleton, $U_i(P,\pi,\tau;\mathfrak{M}) = \{\varnothing\}$, so $|U_i(P,\pi,\tau;\mathfrak{M})| = 1$. Define the maximum local domain size by $q(P) = \max_{v\in V(P)} |D_v|$; since every local domain is nonempty, $q(P) \geq 1$.

3.2 Raw Interface-State Bound

Theorem 3.1 (Raw Interface-State Bound)**.** *For every stage i, $|U_i(P,\pi,\tau;\mathfrak{M})| \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$, and consequently $|U_i(P,\pi,\tau;\mathfrak{M})| \leq q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}$.*

Proof. By definition, $U_i(P,\pi,\tau;\mathfrak{M}) = \prod_{v\in B_i^{\pi,\tau}} D_v$, so $|U_i(P,\pi,\tau;\mathfrak{M})| = \prod_{v\in B_i^{\pi,\tau}} |D_v|$. Every factor satisfies $|D_v| \leq q(P)$, hence $|U_i(P,\pi,\tau;\mathfrak{M})| \leq q(P)^{|B_i^{\pi,\tau}|}$. Because the selected interface is cardinality-minimal, $|B_i^{\pi,\tau}| = b_i^{\pi,\tau}(P) \leq \widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})$; since $q(P) \geq 1$, $q(P)^{|B_i^{\pi,\tau}|} \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$, which proves the first inequality. By the Interface Bound of Chapter 2, $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \leq \omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) + 1$, which gives the second. □

The bound using $\widehat{\omega}_{\mathrm{rec}}$ is the sharper operational form; the normalized expression with $\omega_{\mathrm{rec}} + 1$ is retained for compatibility with the normalized width convention.

3.3 Reachable Interface States

The reachable prefix-state space $\mathcal{R}_i^{\pi}(P) \subseteq \prod_{v\in V_i^{\pi}} D_v$ was defined in Chapter 2. The active-interface projection is $\rho_i^{\pi,\tau} : \mathcal{R}_i^{\pi}(P) \to U_i(P,\pi,\tau;\mathfrak{M})$, $\rho_i^{\pi,\tau}(r) = r|_{B_i^{\pi,\tau}}$.

Definition 3.2 (Reachable Interface-State Space)**.** The reachable interface-state space at stage i is the image $U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) = \rho_i^{\pi,\tau}(\mathcal{R}_i^{\pi}(P)) = \{r|_{B_i^{\pi,\tau}} : r \in \mathcal{R}_i^{\pi}(P)\}$.

Thus $U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \subseteq U_i(P,\pi,\tau;\mathfrak{M})$. Raw and reachable interface states are different objects. **Example.** If a three-component Boolean interface has the raw space $U_i = \{000, 001, 010, 011, 100, 101, 110, 111\}$, the declared transition and admissibility rules

may yield only $U_i^{\mathrm{reach}} = \{001, 011, 101, 111\}$: the first set contains every syntactically possible interface assignment, while the second contains only interface assignments that arise from reachable complete prefix states.

Corollary 3.1 (Reachable Interface-State Bound)**.** *For every stage i, $|U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})| \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$, and consequently $|U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})| \leq q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}$.*

Proof. Since $U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M}) \subseteq U_i(P, \pi, \tau; \mathfrak{M})$, $|U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})| \leq |U_i(P, \pi, \tau; \mathfrak{M})|$; the result follows from the Raw Interface-State Bound (Theorem 3.1). □

3.4 Induced Interface Signatures

Chapter 2 defined the reconstruction-safe prefix-signature map $\mathrm{Sig}_{\tau,i}^{\pi} : \mathcal{R}_i^{\pi}(P) \to \Sigma_{\tau,i}^{\pi}$. Because $B_i^{\pi,\tau}$ is task-sufficient, this map factors through the active-interface projection.

Proposition 3.1 (Induced Interface Signature)**.** *There exists a unique map $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi} : U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M}) \to \Sigma_{\tau,i}^{\pi}$ such that $\mathrm{Sig}_{\tau,i}^{\pi} = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi} \circ \rho_i^{\pi,\tau}$. Equivalently, for $u \in U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})$ one may define $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(u) = \mathrm{Sig}_{\tau,i}^{\pi}(r)$ for any reachable prefix state $r \in \mathcal{R}_i^{\pi}(P)$ satisfying $\rho_i^{\pi,\tau}(r) = u$.*

Proof. Existence and uniqueness follow directly from the Interface Factorization Proposition of Chapter 2. If $\rho_i^{\pi,\tau}(r) = \rho_i^{\pi,\tau}(r') = u$, then $r|_{B_i^{\pi,\tau}} = r'|_{B_i^{\pi,\tau}}$, and since $B_i^{\pi,\tau}$ is task-sufficient, $\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r')$; hence the value assigned to u is independent of the selected prefix-state representative. □

This construction avoids defining a future-continuation set directly for an interface assignment. In general, several reachable prefix states may project to the same interface state while possessing different literal continuation sets; what is guaranteed to be representative-independent is the induced reconstruction-safe signature.

3.5 Task-Relative Semantic Equivalence

Definition 3.3 (Reconstruction-Safe Semantic Equivalence)**.** For $u, v \in U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})$, define $u \equiv_{P,\pi,i}^{\tau,\mathfrak{M}} v$ if and only if $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(u) = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(v)$.

When the problem, task, ordering, and model are fixed, the shorter notation $u \equiv_i^{\tau} v$ or $u \equiv_i v$ may be used. This is the primary semantic-equivalence relation of CPR-WIDTH; it is not defined merely by equality of the current decision, counting, or optimization answer, but incorporates the reconstruction-safe information required by conditions S1–S3 of Chapter 2.

Proposition 3.2 (Equivalence Relation)**.** *For every fixed P, τ, $\mathfrak{M}$, π, and stage i, the relation $\equiv_{P,\pi,i}^{\tau,\mathfrak{M}}$ is an equivalence relation on $U_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})$.*

Proof. Reflexivity and symmetry follow directly from equality of $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}$-values. For transitivity, if $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(u) = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(v)$ and $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(v) = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(w)$, then $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(u) = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(w)$. □

Remark 3.1 (Task-Answer Equality Is Not the Primary Quotient)**.** For a decision task, two prefix states may both satisfy $E_i^{\pi}(r) \neq \varnothing$ while admitting different next values and different successor behavior. For example, $E_i^{\pi}(r) = \{00\}$ and $E_i^{\pi}(r') = \{11\}$ produce the same current decision value $\mathbf{1}[E_i^{\pi}(r) \neq \varnothing] = \mathbf{1}[E_i^{\pi}(r') \neq \varnothing] = 1$, yet the two states may require incompatible next reconstruction choices. Therefore, equality of the current task answer is generally too coarse to define a dynamic reconstruction quotient; the reconstruction-safe signature additionally preserves admissible next values and successor classes.

3.6 Semantic Quotient

Definition 3.4 (Semantic Quotient State Space)**.** The semantic state space at stage i is the quotient $S_i(P,\pi,\tau;\mathfrak{M}) = U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})/\equiv_{P,\pi,i}^{\tau,\mathfrak{M}}$.

An element of $S_i(P,\pi,\tau;\mathfrak{M})$ is an equivalence class $[u]_i^{\tau,\mathfrak{M}} = \{v \in U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) : v \equiv_{P,\pi,i}^{\tau,\mathfrak{M}} u\}$. The canonical quotient map $q_i^{\tau,\mathfrak{M}} : U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \twoheadrightarrow S_i(P,\pi,\tau;\mathfrak{M})$, $q_i^{\tau,\mathfrak{M}}(u) = [u]_i^{\tau,\mathfrak{M}}$, is surjective by construction, so $|S_i(P,\pi,\tau;\mathfrak{M})| \le |U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})|$.

Definition 3.5 (Maximum Semantic-State Count)**.** The maximum semantic-state count is $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = \max_{0\le i\le n} |S_i(P,\pi,\tau;\mathfrak{M})|$.

When the task and model are fixed, one may write $S_i(P,\pi)$ and $S_{\mathrm{CPR}}(P,\pi)$. The symbol $S_i(P,\pi,\tau;\mathfrak{M})$ denotes a quotient set, while $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})$ denotes a number.

3.7 Quotient Transition

To use semantic classes dynamically, stage-to-stage transitions must be well defined on quotient states. Let $u \in U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})$ and choose any reachable prefix representative $r \in \mathcal{R}_i^{\pi}(P)$ satisfying $\rho_i^{\pi,\tau}(r) = u$. Define the admissible next-value set of the interface state by $\widehat{\Lambda}_i^{\pi,\tau}(u) = \Lambda_i^{\pi}(r)$.

Lemma 3.1 (Well-Defined Interface Admissibility)**.** *The set $\widehat{\Lambda}_i^{\pi,\tau}(u)$ is independent of the selected prefix representative r.*

Proof. Let $r, r' \in \mathcal{R}_i^{\pi}(P)$ satisfy $\rho_i^{\pi,\tau}(r) = \rho_i^{\pi,\tau}(r') = u$. Task sufficiency implies $\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r')$; condition S2 of Chapter 2 gives $\Lambda_i^{\pi}(r) = \Lambda_i^{\pi}(r')$. □

Definition 3.6 (Semantic Quotient Transition)**.** For a semantic class $[u]_i^{\tau,\mathfrak{M}} \in S_i(P,\pi,\tau;\mathfrak{M})$ and $a \in \widehat{\Lambda}_i^{\pi,\tau}(u)$, define

$$\overline{\delta}_i^{\pi,\tau}\left([u]_i^{\tau,\mathfrak{M}}, a\right) = \left[\rho_{i+1}^{\pi,\tau}(\delta_i^{\pi}(r,a))\right]_{i+1}^{\tau,\mathfrak{M}},$$

where r is any reachable prefix representative satisfying $\rho_i^{\pi,\tau}(r) = u$.

Proposition 3.3 (Well-Defined Quotient Transition)**.** *The semantic quotient transition $\overline{\delta}_i^{\pi,\tau}$ is independent of the selected interface representative u of the semantic class and of the selected prefix-state representative r of u.*

Proof. Let $[u]_i^{\tau,\mathfrak{M}} = [v]_i^{\tau,\mathfrak{M}}$, so $u \equiv_{P,\pi,i}^{\tau,\mathfrak{M}} v$ and $\widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(u) = \widehat{\mathrm{Sig}}_{\tau,i}^{\pi}(v)$. Choose prefix representatives r, r' with $\rho_i^{\pi,\tau}(r) = u$ and $\rho_i^{\pi,\tau}(r') = v$; by the induced-signature definition, $\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r')$. Condition S2 gives $\Lambda_i^{\pi}(r) = \Lambda_i^{\pi}(r')$, so the same next value a is admissible from both representatives. Condition S3 gives $\mathrm{Sig}_{\tau,i+1}^{\pi}(\delta_i^{\pi}(r,a)) = \mathrm{Sig}_{\tau,i+1}^{\pi}(\delta_i^{\pi}(r',a))$. Applying the induced interface-signature factorization at stage $i+1$ shows the two projected successors belong to the same semantic class at stage $i+1$; hence the quotient transition is well defined. □

3.8 Width–State Bound

Theorem 3.2 (Width–State Bound)**.** *For every finite problem instance P, declared task τ, reconstruction model $\mathfrak{M}$, and admissible ordering π, $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$, and consequently $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}$.*

Proof. For every stage i, $|S_i(P,\pi,\tau;\mathfrak{M})| \le |U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})| \le q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$ by the Reachable Interface-State Bound. Taking the maximum over all reconstruction stages gives $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = \max_{0\le i\le n} |S_i(P,\pi,\tau;\mathfrak{M})| \le q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$. The normalized form follows from $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \le \omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) + 1$. □

This theorem is a cardinality bound. It does not imply that the semantic quotient can be constructed efficiently, nor that the complete reconstruction runtime is polynomial. The Width–State Bound is a worst-case bound and need not be tight, because reachability restrictions and semantic quotienting may reduce the actual semantic-state count substantially below the raw width-based upper bound.

3.9 Binary Semantic-State Dimension

Definition 3.7 (Binary Semantic-State Dimension)**.** The binary semantic-state dimension is

$$D_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = \lceil \log_2 \max\{1, S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})\} \rceil .$$

This quantity measures the number of bits required to identify one semantic state among the maximum number of simultaneously relevant semantic classes; it does not measure the cost of constructing the semantic quotient, the total number of transitions, the cost of reconstructing outputs, or the complete runtime. If $q(P) \geq 2$, the Width–State Bound implies $D_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \leq \lceil \widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \log_2 q(P) \rceil \leq \lceil (\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) + 1) \log_2 q(P) \rceil$. If $q(P) = 1$, every component domain is a singleton, so every raw interface-state space is a singleton, $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = 1$, and $D_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = 0$.

3.10 Task Dependence

The semantic quotient is task-relative. Different tasks may require different reconstruction-safe signature families, even for the same problem representation and reconstruction ordering: for decision, the required final answer may be a Boolean value; for counting, exact multiplicities must be preserved; for enumeration, every required complete output must remain reconstructible; for optimization, the preserved information may include an optimum value, optimal witnesses, or certified residual objective information. Therefore $\equiv^{\mathrm{dec},\mathfrak{M}}_{P,\pi,i}$, $\equiv^{\mathrm{count},\mathfrak{M}}_{P,\pi,i}$, $\equiv^{\mathrm{enum},\mathfrak{M}}_{P,\pi,i}$, and $\equiv^{\mathrm{opt},\mathfrak{M}}_{P,\pi,i}$ need not coincide, and accordingly $S_{\mathrm{CPR}}(P,\pi,\mathrm{dec};\mathfrak{M})$ may differ from $S_{\mathrm{CPR}}(P,\pi,\mathrm{count};\mathfrak{M})$, and similarly for enumeration and optimization. No general pairwise inequality is claimed; for a particular instance, two or more of these task-relative equivalence relations may coincide accidentally.

3.11 Semantic Safety

The former statement that equivalent states preserve every individual future continuation is stronger than required and does not follow from task-relative equality. The correct safety statement concerns the properties guaranteed by the reconstruction-safe signature.

Proposition 3.4 (Semantic Safety)**.** *Let $u, v \in U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})$ and suppose $u \equiv^{\tau,\mathfrak{M}}_{P,\pi,i} v$. Then the declared task-relative future values coincide, the admissible next-value sets coincide, for every admissible next value the corresponding successors belong to the same semantic class at stage $i+1$, and the quotient transition is independent of the selected representative.*

Proof. By semantic equivalence, $\widehat{\mathrm{Sig}}^{\pi}_{\tau,i}(u) = \widehat{\mathrm{Sig}}^{\pi}_{\tau,i}(v)$. Choose reachable prefix representatives r, r' with $\rho_i^{\pi,\tau}(r) = u$ and $\rho_i^{\pi,\tau}(r') = v$; the induced-signature definition gives $\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r')$. Condition S1 of Chapter 2 gives $\mathrm{Ans}^{\pi}_{\tau,i}(r) = \mathrm{Ans}^{\pi}_{\tau,i}(r')$, proving task preservation. Condition S2 gives $\Lambda^{\pi}_i(r) = \Lambda^{\pi}_i(r')$, so the admissible next-value sets coincide. For every common admissible next value a, condition S3 gives $\mathrm{Sig}^{\pi}_{\tau,i+1}(\delta^{\pi}_i(r,a)) = \mathrm{Sig}^{\pi}_{\tau,i+1}(\delta^{\pi}_i(r',a))$, so the projected successors satisfy $\rho^{\pi,\tau}_{i+1}(\delta^{\pi}_i(r,a)) \equiv^{\tau,\mathfrak{M}}_{P,\pi,i+1} \rho^{\pi,\tau}_{i+1}(\delta^{\pi}_i(r',a))$; the quotient transition is consequently independent of the selected representative. □

Semantic safety does not generally imply $E^{\pi}_i(r) = E^{\pi}_i(r')$. For decision, counting, or optimization, different literal continuation sets may still induce the same reconstruction-safe quotient behavior. If exact equality of every continuation is required by the declared task, that stronger requirement must be included explicitly in the signature.

3.12 Structural Interpretation

The state spaces introduced in this chapter satisfy $U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \subseteq U_i(P,\pi,\tau;\mathfrak{M})$, and the semantic quotient is obtained through the surjective map $q_i^{\tau,\mathfrak{M}} : U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \twoheadrightarrow S_i(P,\pi,\tau;\mathfrak{M})$. Therefore $|S_i(P,\pi,\tau;\mathfrak{M})| \le |U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})| \le |U_i(P,\pi,\tau;\mathfrak{M})|$, and the complete stage-wise construction is

$$\mathcal{R}_i^{\pi}(P) \xrightarrow{\rho_i^{\pi,\tau}} U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \xrightarrow{q_i^{\tau,\mathfrak{M}}} S_i(P,\pi,\tau;\mathfrak{M}).$$

The first map forgets processed components outside the task-sufficient active interface; the second merges reachable interface assignments with identical reconstruction-safe signatures. Because the signature satisfies S1–S3, semantic quotienting preserves the declared task-relative future value, the admissible next-value set, and the semantic class of every admissible successor. Figure 3.1 illustrates this construction, from raw and reachable interface states through the semantic quotient to the maximum semantic-state count.

The resulting width–state relation, using either width form, is $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \to S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})$ or equivalently $\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \to S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})$. The present chapter establishes only a structural cardinality bound; the following chapter extends the framework to the complete operation-count model,

$$\boxed{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \longrightarrow S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \longrightarrow N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})}.$$

A small semantic-state count does not by itself imply that semantic classes are inexpensive to construct: a complete runtime conclusion must additionally include construction, reduction, transition, reconstruction, verification, and output costs.

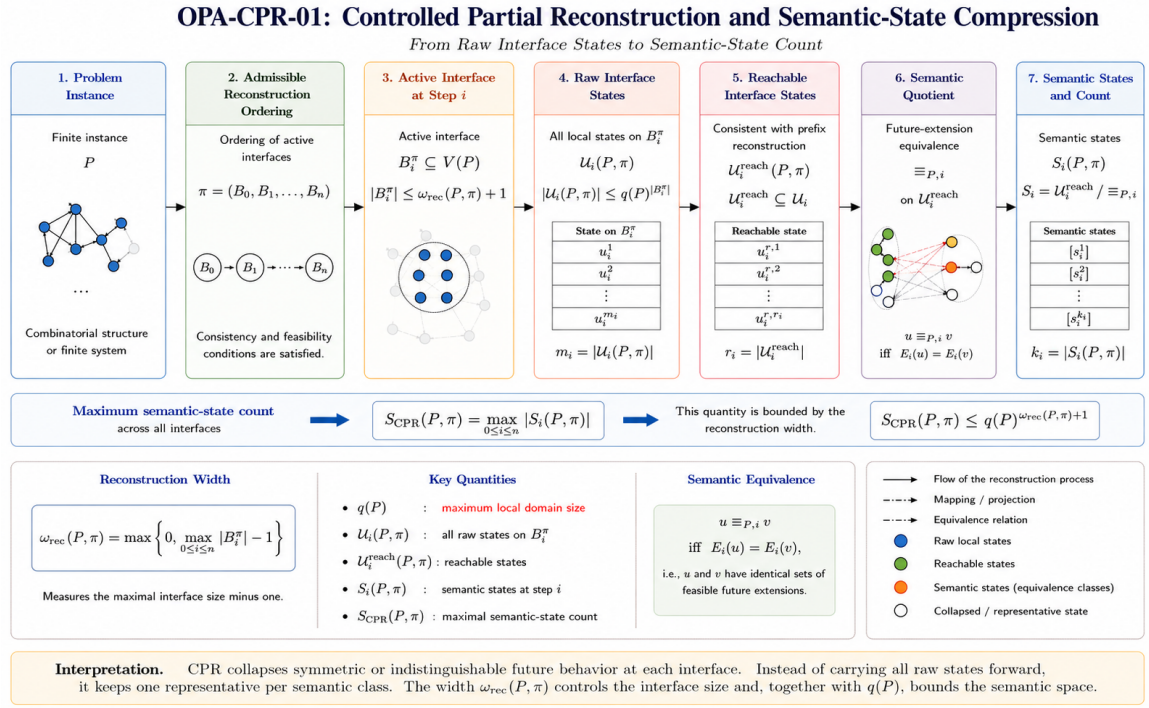


Figure 3.1: OPA-CPR-01: Controlled partial reconstruction and semantic-state compression. The figure summarizes raw interface states $U_i(P,\pi)$, reachable interface states $U_i^{\mathrm{reach}}(P,\pi)$, the semantic quotient $S_i(P,\pi)$, and the maximum semantic-state count $S_{\mathrm{CPR}}(P,\pi)$.

4 Complete Runtime Framework

Controlled Partial Reconstruction Width is a structural parameter. A small active reconstruction interface does not by itself establish a small runtime. A complete computational analysis must additionally account for the costs of constructing the declared reconstruction model, forming active interfaces and reduced representations, generating and processing semantic transitions, reconstructing task-relevant outputs, verifying correctness, and materializing the required output. This chapter introduces the complete operation-count model associated with the CPR-WIDTH framework.

Throughout, let P be a finite problem instance, let τ be the declared computational task, let $\mathfrak{M}$ be the fixed CPR reconstruction model, and let $\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})$ be an admissible reconstruction ordering; all runtime quantities are relative to $(P, \pi, \tau; \mathfrak{M})$, and this dependence may be suppressed when unambiguous. The runtime framework developed here is an accounting model: it specifies which elementary operations must be counted and how the complete operation count is decomposed. It is not, by itself, a hardware-specific implementation model or a measured wall-clock model.

4.1 Complete Cost Model

Definition 4.1 (Complete CPR Operation Count)**.** The complete CPR operation count is

$$N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}, \tag{4.1}$$

all terms evaluated at $(P, \pi, \tau; \mathfrak{M})$.

The six terms are disjoint accounting categories: every elementary operation of the declared CPR procedure must be assigned to exactly one category, never two. In particular, construction of a semantic class belongs either to N_{Reduce} or to $N_{\mathrm{Transition}}$ according to the declared accounting rule, but not both; writing an output object belongs to N_{Output}, not again to $N_{\mathrm{Reconstruct}}$; and verification steps belong to N_{Verify}, unless correctness follows directly from a proved construction with no separate verification operation executed. The accounting convention must be fixed before any count-level comparison is performed.

4.2 Build and Reduction Cost

Definition 4.2 (Build Cost)**.** The build cost $N_{\mathrm{Build}}(P, \pi, \tau; \mathfrak{M})$ is the number of elementary operations required to construct the initial CPR representation before semantic-state processing begins.

Depending on the declared model, this may include parsing and validating the encoded instance, constructing $V(P)$ and the local domains D_v, constructing or receiving the admissible ordering π, evaluating $\mathrm{Adm}_{P,\tau}(\pi)$, initializing reachable-prefix representations, constructing auxiliary indices or constraint tables, and allocating data structures. It does not include later semantic quotienting or stage-to-stage transitions. A favorable ordering is computationally useful only if it is either provided as part of the input or constructible within the declared cost bound.

Definition 4.3 (Reduction Cost)**.** The reduction cost $N_{\mathrm{Reduce}}(P, \pi, \tau; \mathfrak{M})$ is the number of elementary operations required to construct the task-sufficient structural reductions used by the CPR model.

This may include determining task-sufficient candidate interfaces, proving or testing interface sufficiency, selecting cardinality-minimal interfaces algorithmically, computing $b_i^{\pi,\tau}(P)$, projecting reachable prefix states onto active interfaces, constructing induced interface signatures, forming semantic equivalence classes, canonicalizing representatives, and constructing static quotient data used in later transitions. The reduction cost must include the work required to construct the semantic quotient: the existence of a small quotient does not imply that the quotient is

inexpensive to compute. If a semantic signature is precomputed once and reused, its construction belongs to N_{Reduce}; if semantic equivalence is tested dynamically during transition processing, the corresponding tests belong to $N_{\text{Transition}}$. The declared accounting rule must state which convention is used.

4.3 Reconstruction Stages and Branching

The reconstruction ordering processes the component set $V(P)$. For the component-by-component reconstruction model of Chapter 2, the number of nonterminal transition stages is $m_{\text{stage}}(P,\pi) = |V(P)|$ (the symbol m_{stage} is used instead of $b(P,\pi)$ to avoid conflict with the minimum interface size $b_i^{\pi,\tau}(P)$). For $u \in U_i^{\text{reach}}(P,\pi,\tau;\mathfrak{M})$, let $\widehat{\Lambda}_i^{\pi,\tau}(u)$ denote the well-defined admissible next-value set of Chapter 3. Define the stage-wise branching number by

$$\lambda_i(P,\pi,\tau;\mathfrak{M}) = \begin{cases} \max\limits_{u \in U_i^{\text{reach}}(P,\pi,\tau;\mathfrak{M})} \left|\widehat{\Lambda}_i^{\pi,\tau}(u)\right|, & U_i^{\text{reach}}(P,\pi,\tau;\mathfrak{M}) \neq \varnothing, \\ 0, & U_i^{\text{reach}}(P,\pi,\tau;\mathfrak{M}) = \varnothing, \end{cases}$$

and the maximum branching number by $\lambda_{\max}(P,\pi,\tau;\mathfrak{M}) = \max_{0 \le i < m_{\text{stage}}(P,\pi)} \lambda_i(P,\pi,\tau;\mathfrak{M})$ (or 0 if $m_{\text{stage}}(P,\pi) = 0$), so both quantities remain well defined in degenerate cases. If every stage is deterministic, then $\lambda_{\max}(P,\pi,\tau;\mathfrak{M}) \le 1$.

4.4 Semantic Transition Cost

A transition is an admissible move from one semantic state at stage i to one semantic state at stage $i+1$; the quotient transition $\overline{\delta}_i^{\pi,\tau}$ was proved well defined in Chapter 3.

Definition 4.4 (Per-Edge Transition Cost)**.** Let $c_{\text{edge}}(P,\pi,\tau;\mathfrak{M})$ be an upper bound on the elementary-operation cost of processing one admissible semantic transition edge: generating a successor candidate, testing local admissibility, evaluating or retrieving its signature, locating or constructing the successor class, merging duplicates, and updating transition data.

Definition 4.5 (Local State-Processing Cost)**.** Let $c_{\text{local}}(P,\pi,\tau;\mathfrak{M})$ be an upper bound on the state-local cost incurred once per semantic state during one reconstruction stage, independently of the number of outgoing transition edges: state initialization, local bookkeeping, retrieval of cached data, and state-level canonicalization.

Definition 4.6 (Per-State Transition Cost)**.** Let $c_{\text{state}}(P,\pi,\tau;\mathfrak{M})$ be an upper bound on the complete cost of processing one semantic state and all its admissible outgoing transitions during one stage.

The per-state and per-edge conventions are related by

$$c_{\text{state}}(P,\pi,\tau;\mathfrak{M}) \;\le\; \lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\text{edge}}(P,\pi,\tau;\mathfrak{M}) \;+\; c_{\text{local}}(P,\pi,\tau;\mathfrak{M}).$$

The manuscript may use either convention, but the selected convention must be stated explicitly.

Definition 4.7 (Transition Cost)**.** The semantic transition cost $N_{\text{Transition}}(P,\pi,\tau;\mathfrak{M})$ is the total number of elementary operations required to process all reachable semantic transitions of the declared CPR procedure.

Using the per-state convention,

$$\begin{aligned} N_{\text{Transition}}(P,\pi,\tau;\mathfrak{M}) &\le \sum_{i=0}^{m_{\text{stage}}(P,\pi)-1} |S_i(P,\pi,\tau;\mathfrak{M})| \cdot c_{\text{state}}(P,\pi,\tau;\mathfrak{M}) \\ &\le m_{\text{stage}}(P,\pi)\, S_{\text{CPR}}(P,\pi,\tau;\mathfrak{M})\, c_{\text{state}}(P,\pi,\tau;\mathfrak{M}) \end{aligned}$$

(the sum is empty, hence 0, if $m_{\text{stage}}(P,\pi) = 0$). Using the per-edge convention, including the local cost incurred once per semantic state,

$$\begin{aligned} N_{\text{Transition}}(P,\pi,\tau;\mathfrak{M}) \le\; & m_{\text{stage}}(P,\pi)\, S_{\text{CPR}}(P,\pi,\tau;\mathfrak{M}) \\ & \times \left[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\text{edge}}(P,\pi,\tau;\mathfrak{M}) + c_{\text{local}}(P,\pi,\tau;\mathfrak{M})\right], \end{aligned}$$

making both the branching contribution and the state-local overhead explicit. (No symbol $\tau(P,\pi)$ is used for transition cost, since τ is reserved for the declared computational task.)

4.5 Width-Based Transition Bound

By the Width–State Bound of Chapter 3, $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$. Consequently, the per-state transition bound becomes

$$N_{\mathrm{Transition}}(P,\pi,\tau;\mathfrak{M}) \le m_{\mathrm{stage}}(P,\pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}\, c_{\mathrm{state}}(P,\pi,\tau;\mathfrak{M}),$$

or, using the normalized width, $m_{\mathrm{stage}}(P,\pi)\, q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}\, c_{\mathrm{state}}(P,\pi,\tau;\mathfrak{M})$. Under the per-edge convention,

$$\begin{aligned} N_{\mathrm{Transition}}(P,\pi,\tau;\mathfrak{M}) \le\ & m_{\mathrm{stage}}(P,\pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \\ & \times \left[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M}) + c_{\mathrm{local}}(P,\pi,\tau;\mathfrak{M})\right], \end{aligned}$$

equivalently with $q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}$ using the normalized width. The bound using $\widehat{\omega}_{\mathrm{rec}}$ is the sharper operational form; the normalized form is retained for consistency with the principal CPR-Width notation.

4.6 Reconstruction, Verification, and Output Cost

Definition 4.8 (Reconstruction Cost)**.** The reconstruction cost $N_{\mathrm{Reconstruct}}(P,\pi,\tau;\mathfrak{M})$ is the number of elementary operations required to recover the task-relevant output from the semantic reconstruction process.

The meaning of this cost depends on the declared task: for decision, it may require only an acceptance value or one witness; for counting, the propagation and recovery of exact multiplicities; for enumeration, the explicit recovery of every required solution; for optimization, an optimum value, an optimal witness, all optimal witnesses, or another declared certificate. It includes operations used to trace predecessor information, expand semantic representatives, recover component values, or assemble complete outputs, but not the physical writing of final outputs, which belongs to N_{Output}.

Definition 4.9 (Verification Cost)**.** The verification cost $N_{\mathrm{Verify}}(P,\pi,\tau;\mathfrak{M})$ is the number of elementary operations required to confirm that the reconstructed result satisfies the declared task conditions.

Verification may include feasibility checks, constraint checks, witness validation, objective-value validation, multiplicity consistency checks, completeness checks for enumeration, or comparison with a certified task value. It may be omitted only when correctness follows directly from the construction and this implication has been proved; merely asserting that an output is valid does not eliminate the verification term unless no separate verification procedure is actually executed.

Definition 4.10 (Output Cost)**.** The output cost $N_{\mathrm{Output}}(P,\pi,\tau;\mathfrak{M})$ is the number of elementary operations required to write, store, transmit, or otherwise materialize the result required by the task.

For a decision task the output cost may be constant; for a witness problem it is at least proportional to the witness length; for enumeration the output may be exponentially large in the input length. Let $L_{\mathrm{Output}}(P,\pi,\tau;\mathfrak{M})$ denote the total encoded output length, and assume an elementary output model in which writing one output symbol or bit requires at least a fixed positive number $c_{\mathrm{out}} > 0$ of elementary operations. Then $N_{\mathrm{Output}}(P,\pi,\tau;\mathfrak{M}) \ge c_{\mathrm{out}}\, L_{\mathrm{Output}}(P,\pi,\tau;\mathfrak{M})$; no runtime statement may ignore this unavoidable output-size lower bound.

For compact notation, define the post-transition cost $R_{\mathrm{post}}(P,\pi,\tau;\mathfrak{M}) = N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$, so that $N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + R_{\mathrm{post}}(P,\pi,\tau;\mathfrak{M})$ is only a notational abbreviation; the six-component decomposition of Definition 4.1 remains the primary

accounting model.

4.7 Complete Width-Based Runtime Bounds

Theorem 4.1 (Complete Width-Based Runtime Bound)**.** *Under the per-state transition-cost convention,*

$$N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + m_{\mathrm{stage}}(P,\pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}\, c_{\mathrm{state}}(P,\pi,\tau;\mathfrak{M}) + R_{\mathrm{post}}(P,\pi,\tau;\mathfrak{M}),$$

and consequently, using the normalized width, $N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + m_{\mathrm{stage}}(P,\pi)\, q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}\, c_{\mathrm{state}}(P,\pi,\tau;\mathfrak{M}) + R_{\mathrm{post}}(P,\pi,\tau;\mathfrak{M})$.

Proof. By the six-component decomposition and the definition of R_{post}, $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + R_{\mathrm{post}}$. The per-state transition bound gives $N_{\mathrm{Transition}} \le m_{\mathrm{stage}}\, S_{\mathrm{CPR}}\, c_{\mathrm{state}}$, and the Width–State Bound of Chapter 3 gives $S_{\mathrm{CPR}} \le q(P)^{\widehat{\omega}_{\mathrm{rec}}}$; substitution proves the first bound. Chapter 3 established $q(P) \ge 1$ and $\widehat{\omega}_{\mathrm{rec}} \le \omega_{\mathrm{rec}} + 1$; since $x \mapsto q(P)^x$ is nondecreasing for $q(P) \ge 1$, it follows that $q(P)^{\widehat{\omega}_{\mathrm{rec}}} \le q(P)^{\omega_{\mathrm{rec}}+1}$, proving the normalized bound. □

Theorem 4.2 (Complete Edge-Based Runtime Bound)**.** *Under the per-edge transition-cost convention,*

$$\begin{aligned} N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le\ & N_{\mathrm{Build}} + N_{\mathrm{Reduce}} \\ & + m_{\mathrm{stage}}(P,\pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \left[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M}) \right. \\ & \qquad \left. + c_{\mathrm{local}}(P,\pi,\tau;\mathfrak{M})\right] \\ & + R_{\mathrm{post}}(P,\pi,\tau;\mathfrak{M}). \end{aligned}$$

Proof. At every stage, at most $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})$ semantic states are processed; each costs at most $c_{\mathrm{local}}(P,\pi,\tau;\mathfrak{M})$ for state-local work and has at most $\lambda_{\max}(P,\pi,\tau;\mathfrak{M})$ admissible outgoing transitions, each costing at most $c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M})$. Therefore $N_{\mathrm{Transition}}(P,\pi,\tau;\mathfrak{M}) \le m_{\mathrm{stage}}(P,\pi)\, S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M}) + c_{\mathrm{local}}(P,\pi,\tau;\mathfrak{M})]$; using $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \le q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$ and substituting into $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + R_{\mathrm{post}}$ proves the bound. □

4.8 Polynomial Tractability Criterion

Let $\ell(P)$ denote the encoding length of the problem instance. A polynomial-time conclusion requires polynomial control of every term in the complete runtime model.

Corollary 4.1 (Constant-Domain Logarithmic-Width Criterion)**.** *Assume that there exists a constant* $q_0 \ge 1$ *such that* $1 \le q(P) \le q_0$*, and that* $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) = O(\log \ell(P))$*. Then there exists a constant* $c > 0$ *such that* $q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \le \ell(P)^c$ *for all sufficiently large inputs; equivalently,* $q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} = \ell(P)^{O(1)}$.

Proof. If $q(P) = 1$, then $q(P)^{\widehat{\omega}_{\mathrm{rec}}} = 1$ for every input, which is trivially polynomially bounded. If $q(P) \ge 2$, since $q(P) \le q_0$ and $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) = O(\log \ell(P))$, there exists a constant $C > 0$ with $\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \le C \log \ell(P)$ for all sufficiently large instances. Therefore $q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \le q_0^{C \log \ell(P)} = \ell(P)^{C \log q_0}$; since q_0 and C are constants, the exponent $C \log q_0$ is constant, so the right-hand side is polynomially bounded in $\ell(P)$. □

This corollary controls only the width-dependent state factor. A complete polynomial-time result additionally requires polynomials $p_{\mathrm{Build}}, p_{\mathrm{Reduce}}, p_{\mathrm{Transition}}, p_{\mathrm{Reconstruct}}, p_{\mathrm{Verify}}, p_{\mathrm{Output}}$ such that $N_j(P,\pi,\tau;\mathfrak{M}) \le p_j(\ell(P))$ for every component j; in particular one must also control $m_{\mathrm{stage}}(P,\pi)$, $\lambda_{\max}(P,\pi,\tau;\mathfrak{M})$, and either $c_{\mathrm{state}}(P,\pi,\tau;\mathfrak{M})$ or $c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M})$ together with $c_{\mathrm{local}}(P,\pi,\tau;\mathfrak{M})$, according to the selected transition-cost convention.

Theorem 4.3 (Sufficient Complete Polynomial-Cost Condition)**.** *Assume there exist polynomials* $p_{\mathrm{Build}}, p_{\mathrm{Reduce}}, p_{\mathrm{Transition}}, p_{\mathrm{Reconstruct}}, p_{\mathrm{Verify}}, p_{\mathrm{Output}}$ *such that* $N_j(P, \pi, \tau; \mathfrak{M}) \le p_j(\ell(P))$ *for every one of the six cost components. Then there exists a polynomial* p_{CPR} *such that* $N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) \le p_{\mathrm{CPR}}(\ell(P))$.

Proof. By Equation (4.1), $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$. Using the assumed bounds, $N_{\mathrm{CPR}} \le p_{\mathrm{Build}}(\ell(P)) + p_{\mathrm{Reduce}}(\ell(P)) + p_{\mathrm{Transition}}(\ell(P)) + p_{\mathrm{Reconstruct}}(\ell(P)) + p_{\mathrm{Verify}}(\ell(P)) + p_{\mathrm{Output}}(\ell(P))$. A finite sum of polynomials is a polynomial, so a polynomial p_{CPR} with this property exists. □

4.9 Classical Baseline, Count-Level Gain, and Wall-Clock Gain

Let $N_{\mathrm{Classic}}(P, \tau; \mathcal{A}_{\mathrm{Classic}})$ denote the complete elementary-operation count of a declared classical baseline algorithm $\mathcal{A}_{\mathrm{Classic}}$ for the same problem representation, declared task, correctness requirement, and output requirement; the baseline must explicitly state the algorithm, the representation, all included preprocessing, the stopping condition, the verification procedure, and the required output. Assume $N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) > 0$.

Definition 4.11 (Count-Level Gain)**.** The count-level gain is an elementary-operation-count metric, not a measured execution-time metric. The count-level gain of the CPR model relative to the declared classical baseline is

$$G_{\mathrm{count}}(P, \pi, \tau; \mathfrak{M}, \mathcal{A}_{\mathrm{Classic}}) = \frac{N_{\mathrm{Classic}}(P, \tau; \mathcal{A}_{\mathrm{Classic}})}{N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})}.$$

A favorable count-level regime exists when $N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) < N_{\mathrm{Classic}}(P, \tau; \mathcal{A}_{\mathrm{Classic}})$, equivalently $G_{\mathrm{count}}(P, \pi, \tau; \mathfrak{M}, \mathcal{A}_{\mathrm{Classic}}) > 1$. A count-level comparison is meaningful only when both methods solve the same declared task and materialize the same required output; a baseline based on exhaustive enumeration is one possible baseline, but not the only admissible one.

Let $T_{\mathrm{Classic}}(P, \tau; \mathcal{I}_{\mathrm{Classic}})$ denote the measured wall-clock time of a declared classical implementation $\mathcal{I}_{\mathrm{Classic}}$, and let $T_{\mathrm{CPR}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}}) > 0$ denote the measured wall-clock time of a declared CPR implementation $\mathcal{I}_{\mathrm{CPR}}$.

Definition 4.12 (Measured Wall-Clock Gain)**.** The measured wall-clock gain is

$$G_{\mathrm{time}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}}, \mathcal{I}_{\mathrm{Classic}}) = \frac{T_{\mathrm{Classic}}(P, \tau; \mathcal{I}_{\mathrm{Classic}})}{T_{\mathrm{CPR}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}})}.$$

A count-level gain does not imply a wall-clock gain, $G_{\mathrm{count}} > 1 \not\Longrightarrow G_{\mathrm{time}} > 1$. Measured runtime may depend on implementation overhead, memory allocation, cache behavior, data movement, parallelism, synchronization, compiler behavior, hardware architecture, operating-system effects, and measurement protocol. Operation-count claims and wall-clock claims must therefore remain separate.

4.10 Structural Runtime Chain and Non-Implications

The theory developed in Chapters 2–4 yields the central structural runtime chain

$$\widehat{\boldsymbol{\omega}}_{\mathbf{rec}}(\boldsymbol{P}, \boldsymbol{\pi}, \boldsymbol{\tau}; \mathfrak{M}) \longrightarrow \boldsymbol{S}_{\mathbf{CPR}}(\boldsymbol{P}, \boldsymbol{\pi}, \boldsymbol{\tau}; \mathfrak{M}) \longrightarrow \boldsymbol{N}_{\mathbf{CPR}}(\boldsymbol{P}, \boldsymbol{\pi}, \boldsymbol{\tau}; \mathfrak{M}),$$

or, using the normalized width, the same chain with ω_{rec} in place of $\widehat{\omega}_{\mathrm{rec}}$. The first arrow is a structural cardinality relation: the active-interface width bounds the raw interface-state space, which bounds the reachable interface-state space, which in turn bounds the semantic quotient. The second arrow is an accounting relation: the semantic-state count contributes to the transition cost, while the remaining cost components determine the complete computational effort. Neither arrow is a universal tractability implication.

Proposition 4.1 (Width Does Not Determine Runtime Alone)**.** *The inequality* $\omega_{\mathrm{rec}}(P, \pi_1, \tau; \mathfrak{M}) < \omega_{\mathrm{rec}}(P, \pi_2, \tau; \mathfrak{M})$ *does not by itself imply* $N_{\mathrm{CPR}}(P, \pi_1, \tau; \mathfrak{M}) < N_{\mathrm{CPR}}(P, \pi_2, \tau; \mathfrak{M})$. *It also does*

not imply $T_{\mathrm{CPR}}(P, \pi_1, \tau; \mathcal{I}_{\mathrm{CPR}}) < T_{\mathrm{CPR}}(P, \pi_2, \tau; \mathcal{I}_{\mathrm{CPR}})$*, even when both orderings are evaluated under the same declared CPR implementation and experimental conditions.*

Proof. A smaller-width ordering may require more expensive ordering construction, more expensive interface-minimality tests, costlier semantic signatures, larger branching, more expensive quotient transitions, greater reconstruction or verification overhead, or less favorable memory behavior. Therefore width alone determines neither the complete operation count nor the measured wall-clock time. □

Proposition 4.2 (Semantic-State Count Does Not Determine Runtime Alone)**.** *The inequality* $S_{\mathrm{CPR}}(P, \pi_1, \tau; \mathfrak{M}) < S_{\mathrm{CPR}}(P, \pi_2, \tau; \mathfrak{M})$ *does not by itself imply* $N_{\mathrm{CPR}}(P, \pi_1, \tau; \mathfrak{M}) < N_{\mathrm{CPR}}(P, \pi_2, \tau; \mathfrak{M})$*.*

Proof. A smaller semantic quotient may be more expensive to construct, and may also require more expensive canonicalization, equivalence tests, or reconstruction steps. The complete operation count depends on all six cost components, not only on the maximum semantic-state count. □

4.11 Conclusion

Controlled Partial Reconstruction Width is a structural control parameter; it does not determine runtime by itself. The complete CPR operation count is

$$N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}},$$

six disjoint accounting categories to which every elementary operation must be assigned exactly once. The transition cost must account explicitly for the number of reconstruction stages, the number of semantic states, the number of admissible outgoing transitions, the state-local processing cost, and the cost per transition edge: under the per-edge convention, $N_{\mathrm{Transition}} \leq m_{\mathrm{stage}}\, S_{\mathrm{CPR}}(\lambda_{\max} c_{\mathrm{edge}} + c_{\mathrm{local}})$; under the per-state convention, $N_{\mathrm{Transition}} \leq m_{\mathrm{stage}}\, S_{\mathrm{CPR}}\, c_{\mathrm{state}}$. The width-based bound controls the semantic-state contribution through $S_{\mathrm{CPR}} \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}} \leq q(P)^{\omega_{\mathrm{rec}}+1}$. A complete polynomial-time conclusion is permitted only when every cost component is polynomially bounded in the encoding length.

The central structural-runtime chain remains $\boldsymbol{\omega_{\mathrm{rec}} \to S_{\mathrm{CPR}} \to N_{\mathrm{CPR}}}$, and a refined runtime-accounting chain is $\boldsymbol{\omega_{\mathrm{rec}} \to S_{\mathrm{CPR}} \to N_{\mathrm{Transition}} \to N_{\mathrm{CPR}}}$, making explicit that the semantic-state count enters the complete operation count primarily through transition processing, without implying that the remaining cost components are determined by the transition cost or by the reconstruction width. Figure 4.1 summarizes the complete pipeline. The following chapter formulates the complete boundary conditions and uniform polynomial-cost requirements needed for CPR tractability.

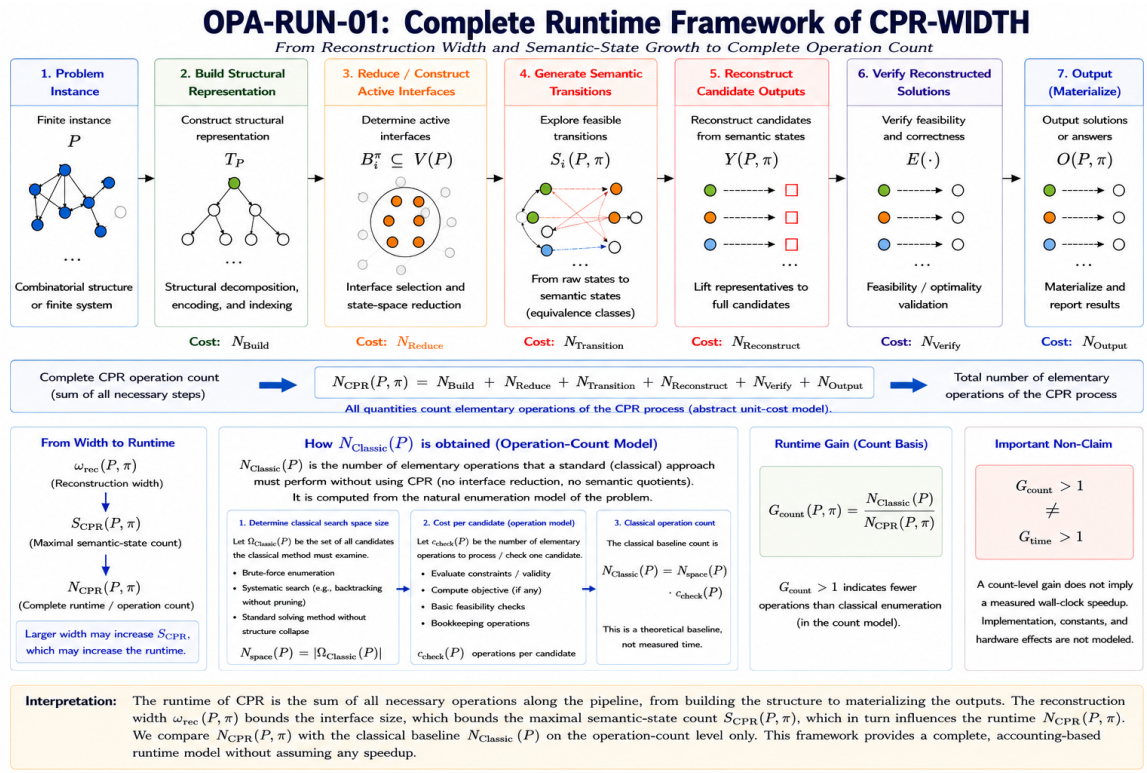


Figure 4.1: OPA-RUN-01: Complete runtime framework of CPR-WIDTH. The figure summarizes the full operation-count model $N_{\text{CPR}}(P,\pi)$, including build, reduce, transition, reconstruct, verify, and output components.

5 Boundary Conditions for CPR Tractability

The structural and runtime framework developed in the preceding chapters does not by itself imply polynomial-time tractability. Additional uniform conditions are required to ensure that Controlled Partial Reconstruction can be executed within polynomial resources for an entire encoded problem family. This chapter formulates those conditions and establishes the resulting tractability criterion.

Throughout, all elementary-operation counts are interpreted in one fixed deterministic bit-cost computation model, which may be taken to be a deterministic multi-tape Turing machine; a deterministic RAM model may be used only when its word size, memory-access rules, and arithmetic-cost conventions are polynomially related to the input encoding length and all required bit-level costs are included in the operation count. No elementary operation may conceal the manipulation of an object whose encoded length is not itself accounted for in the complete runtime model.

Let $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$ be a uniformly encoded family of finite problem instances, where x is the binary encoding of P_x, and let $\ell(P_x) = |x|$ denote the encoding length. Let τ be the declared computational task, let $\mathfrak{M}$ be a uniform CPR reconstruction model for the family, and let $\pi_x \in \Pi_{\mathrm{adm}}(P_x, \tau; \mathfrak{M})$ be the admissible reconstruction ordering used for the encoded instance P_x.

5.1 Overview

Polynomial-time tractability requires more than a small reconstruction width. The reconstruction representation must be explicitly specified and uniformly constructible; the semantic reconstruction process must be sound and complete for the declared task; the reachable semantic-state system and its transition structure must be polynomially controllable; and every component of the complete runtime model of Chapter 4 must be polynomially bounded. These requirements are organized through four boundary conditions: BC1, explicit uniform structural representation; BC2, polynomial uniform constructibility; BC3, task-relative soundness and completeness; BC4, polynomial semantic-state and transition-structure control. The boundary conditions are necessary components of the declared CPR tractability certificate; they do not, considered in isolation, replace the complete six-component runtime analysis.

5.2 Boundary Condition BC1

Definition 5.1 (BC1 – Explicit Uniform Structural Representation)**.** A uniformly encoded family $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$ satisfies BC1 for task τ and reconstruction model $\mathfrak{M}$ if the model explicitly specifies, for every encoded instance P_x: the finite reconstruction-component representation $V(P_x)$ and local domains D_v; the admissibility predicate for reconstruction orderings; the representation of an admissible ordering when supplied as part of the instance, or one uniform procedure for constructing one; the reachable prefix-state spaces; the task-sufficient active reconstruction interfaces; the reconstruction-safe signature family $\mathrm{Sig}_{\tau,i}^{\pi_x}$; the reachable interface-state spaces; the semantic equivalence relation and semantic quotient; the quotient transition rule; the reconstruction operator; the verification rule; and the required output representation. All objects must be specified by one uniform finite description that applies to the complete encoded family.

BC1 is a representation condition: it requires the mathematical objects underlying the CPR analysis to be explicitly declared. It does not assert that these objects are inexpensive to construct, nor that the resulting reconstruction process is sound, complete, or polynomially bounded.

5.3 Boundary Condition BC2

Definition 5.2 (BC2 – Polynomial Uniform Constructibility)**.** A uniformly encoded family $\mathcal{F}$ satisfies BC2 for task τ and model $\mathfrak{M}$ if there exists one deterministic preprocessing algorithm $\mathcal{A}_{\text{Preprocess}}$ and one polynomial $p_{\text{BC2}} : \mathbb{N} \to \mathbb{N}$ such that, for every encoding x, $N_{\mathcal{A}_{\text{Preprocess}}}(x) \leq p_{\text{BC2}}(|x|)$, where $N_{\mathcal{A}_{\text{Preprocess}}}(x)$ is the number of elementary operations executed by the preprocessing algorithm, which constructs or reads all structural data required before dynamic semantic-state processing begins.

The constructed or supplied data include, whenever applicable, the finite component representation, the local-domain representation, an admissible reconstruction ordering, the task-sufficient active-interface representation, the static reconstruction-safe signature data, auxiliary incidence and indexing structures, and all other data assigned to N_{Build} or the static portion of N_{Reduce}. The ordering-supply procedure is part of $\mathcal{A}_{\text{Preprocess}}$: it either reads an admissible ordering from the declared input or constructs one uniformly from x; in both cases $\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M})$, and the operation count of $\mathcal{A}_\pi$ is included in N_{Build}, not counted as an additional runtime component. The quantity $N_{\mathcal{A}_{\text{Preprocess}}}$ is an auxiliary preprocessing count and does not introduce a seventh component into the complete runtime model; every elementary preprocessing operation is assigned either to N_{Build} or to N_{Reduce}, following the disjoint accounting convention of Chapter 4, but not to both.

BC2 requires polynomial constructibility of every structural object the declared model assigns to preprocessing, and is uniform: the same algorithm and the same polynomial bound must apply to every instance in the encoded family; an instance-specific construction procedure or an instance-dependent polynomial is not sufficient. BC2 does not by itself imply that semantic transitions, reconstruction, verification, or output materialization are polynomially bounded; those costs remain separate components of the complete runtime model.

5.4 Boundary Condition BC3

Definition 5.3 (BC3 – Task-Relative Soundness and Completeness)**.** A CPR reconstruction model satisfies BC3 for a declared task τ if its reconstructed result is both sound and complete with respect to the exact output requirement of τ.

The precise condition depends on the declared task. For every task τ, let $\text{Sol}_\tau(P_x)$ denote the complete set of feasible objects relevant to that task, whenever such a representation is applicable.

For **decision**, let $\text{Dec}_\tau(P_x) \in \{0, 1\}$ denote the correct task-relative decision value and $\text{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) \in \{0, 1\}$ the CPR-computed value; soundness and completeness require $\text{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = \text{Dec}_\tau(P_x)$, which for an existential feasibility decision task specializes to $\text{Sol}_\tau(P_x) \neq \varnothing \Leftrightarrow \text{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = 1$. The CPR procedure must return the correct yes-or-no answer; it need not preserve every individual feasible witness unless witness reconstruction is part of the declared task.

For **enumeration**, soundness and completeness require $\text{Output}_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) = \text{Sol}_\tau(P_x)$, equivalently soundness $\text{Output}_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) \subseteq \text{Sol}_\tau(P_x)$ and completeness $\text{Sol}_\tau(P_x) \subseteq \text{Output}_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M})$.

For **counting**, soundness and completeness require $\text{CPRCount}(P_x, \pi_x, \tau; \mathfrak{M}) = |\,\text{Sol}_\tau(P_x)|$.

For **optimization**, let $\text{Opt}_\tau(P_x)$ denote the exact task-required optimum object, which may be the optimum value, one optimal witness, all optimal witnesses, or another explicitly declared optimality certificate; BC3 requires $\text{CPROpt}(P_x, \pi_x, \tau; \mathfrak{M}) = \text{Opt}_\tau(P_x)$. No weaker output object may silently replace the declared task requirement.

More generally, if the task is represented by the evaluation map Θ_τ, BC3 requires exact preservation of the declared task value, $\Theta_\tau^{\text{CPR}}(P_x, \pi_x; \mathfrak{M}) = \Theta_\tau(P_x)$; the decision, enumeration, counting, and optimization conditions above are special cases of this identity. BC3 is a correctness condition and does not by itself provide a polynomial runtime bound.

5.5 Boundary Condition BC4

For each encoded instance, define the total number of semantic-state occurrences by $V_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M}) = \sum_{i=0}^{m_{\mathrm{stage}}(P_x,\pi_x)} |S_i(P_x, \pi_x, \tau; \mathfrak{M})|$; the same abstract state occurring at two different stages is counted twice, since the two occurrences belong to different stage-specific semantic-state spaces. Let $E_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M})$ denote the total number of reachable semantic transition edges processed by the declared CPR reconstruction system.

Definition 5.4 (BC4 – Polynomial Semantic-State and Transition-Structure Control)**.** A uniformly encoded family $\mathcal{F}$ satisfies BC4 for task τ, model $\mathfrak{M}$, and admissible orderings π_x if there exist polynomials $p_{\mathrm{SemanticVertex}}, p_{\mathrm{SemanticEdge}} : \mathbb{N} \to \mathbb{N}$, independent of x, such that for every encoded instance P_x, $V_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_{\mathrm{SemanticVertex}}(|x|)$ and $E_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_{\mathrm{SemanticEdge}}(|x|)$.

The first inequality controls the complete number of stage-indexed reachable semantic states, and the second controls the complete number of reachable semantic transition edges. Since $S_{\mathrm{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) = \max_i |S_i(P_x, \pi_x, \tau; \mathfrak{M})| \leq V_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M})$, BC4 also implies polynomial control of the maximum stage-wise semantic-state count. A short binary description of one semantic state does not imply that only polynomially many semantic states occur, and polynomially many semantic states do not imply polynomially many semantic transition edges when the branching structure is not polynomially controlled.

BC4 is a structural cardinality condition; it does not by itself imply that processing one transition edge is inexpensive. The complete elementary-operation cost of generating, identifying, merging, storing, and processing the semantic transitions remains $N_{\mathrm{Transition}}(P_x, \pi_x, \tau; \mathfrak{M})$, which must be controlled separately by the complete runtime model, and BC4 also does not eliminate the need to control reconstruction, verification, and output costs separately. BC4 is retained as an independent structural certification condition, although the complete polynomial-time conclusion is ultimately obtained from the six component-wise operation-count bounds.

5.6 CPR-Tractable Families and the Language Class

The expression "CPR-tractable" is used only when the four boundary conditions and the complete polynomial-cost requirements are satisfied uniformly.

Definition 5.5 (CPR-Tractable Decision Family)**.** Let $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$ be a uniformly encoded family of finite decision-problem instances. The family is CPR-tractable if there exist one declared decision task τ, one uniform CPR reconstruction model $\mathfrak{M}$, one uniform deterministic preprocessing algorithm $\mathcal{A}_{\mathrm{Preprocess}}$, containing an ordering-supply component $\mathcal{A}_\pi$ with $\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\mathrm{adm}}(P_x, \tau; \mathfrak{M})$, and six fixed polynomial functions $p_j : \mathbb{N} \to \mathbb{N}$, $j \in J = \{\mathrm{Build}, \mathrm{Reduce}, \mathrm{Transition}, \mathrm{Reconstruct}, \mathrm{Verify}, \mathrm{Output}\}$, such that BC1–BC4 hold uniformly and, for every encoding x, $N_j(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_j(|x|)$ for every $j \in J$.

The notation $\mathcal{A}_\pi(x)$ also covers the case in which the ordering is explicitly included in the declared input representation and is read and validated by the ordering-supply component. Let the decision language associated with the family be $L_{\mathcal{F}} = \{x \in \{0,1\}^* : P_x \text{ is a yes-instance for the declared task } \tau\}$.

Definition 5.6 (The Language Class $\mathbf{P}_{\mathrm{CPR}}$)**.** Define $\mathbf{P}_{\mathrm{CPR}} = \{L \subseteq \{0,1\}^* : L = L_{\mathcal{F}} \text{ for some uniformly encoded CPR-tractable decision family } \mathcal{F}\}$.

Membership in $\mathbf{P}_{\mathrm{CPR}}$ therefore requires more than satisfaction of BC1–BC4 as isolated statements: it requires a uniform CPR decision procedure with polynomial bounds for all six components of the complete operation-count model. The class is framework-dependent because certification depends on the declared component representation, the task, the reconstruction model, the admissible-ordering rule, the reconstruction-safe semantic signatures, and the complete runtime-accounting convention.

5.7 Uniform Tractability

Theorem 5.1 (Uniform CPR Tractability)**.** *Let $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$ be a uniformly encoded family of finite decision-problem instances with associated decision language $L_{\mathcal{F}}$. Assume there exist one declared decision task τ, one uniform CPR reconstruction model $\mathfrak{M}$, one uniform deterministic preprocessing algorithm $\mathcal{A}_{\text{Preprocess}}$ with ordering-supply component $\mathcal{A}_\pi$ producing $\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M})$, and six fixed polynomial functions $p_j : \mathbb{N} \to \mathbb{N}$, $j \in J$, such that BC1, BC2, BC3, and BC4 hold uniformly and, for every encoding x, $N_j(P_x, \pi_x, \tau; \mathfrak{M}) \le p_j(|x|)$ for every $j \in J$. Then $L_{\mathcal{F}} \in \mathbf{P}$.*

Proof. By BC1, the complete CPR decision model is explicitly and uniformly specified. By BC2, the reconstruction representation, all required structural data, and an admissible ordering are read or constructed using a number of elementary operations polynomial in $|x|$. By BC3, the CPR decision procedure is sound and complete: $x \in L_{\mathcal{F}} \Leftrightarrow \text{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = 1$. By BC4, the complete stage-indexed semantic-state system and its semantic transition graph have polynomially bounded cardinality; the complete cost of processing that graph is controlled separately by the assumed polynomial bound on $N_{\text{Transition}}$. By the complete operation-count decomposition of Chapter 4, $N_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) = N_{\text{Build}} + N_{\text{Reduce}} + N_{\text{Transition}} + N_{\text{Reconstruct}} + N_{\text{Verify}} + N_{\text{Output}}$; using the six assumed uniform polynomial bounds gives $N_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) \le p_{\text{Build}}(|x|) + p_{\text{Reduce}}(|x|) + p_{\text{Transition}}(|x|) + p_{\text{Reconstruct}}(|x|) + p_{\text{Verify}}(|x|) + p_{\text{Output}}(|x|)$, a finite sum of fixed polynomials and hence itself polynomially bounded in $|x|$. Because the operation counts are interpreted in the fixed deterministic bit-cost model declared at the start of this chapter, the complete CPR procedure is executable in deterministic polynomial time, so membership of x in $L_{\mathcal{F}}$ is decided by one deterministic polynomial-time procedure. Hence $L_{\mathcal{F}} \in \mathbf{P}$. □

Corollary 5.1 (Certified CPR Languages Are Polynomial-Time Languages)**.** $\mathbf{P}_{\text{CPR}} \subseteq \mathbf{P}$.

Proof. Let $L \in \mathbf{P}_{\text{CPR}}$. By Definition 5.6, L is associated with a uniformly encoded CPR-tractable decision family, so the hypotheses of Theorem 5.1 hold, and hence $L \in \mathbf{P}$. □

This inclusion is a consequence of the certification requirements built into the definition of $\mathbf{P}_{\text{CPR}}$; it is not a complexity-class collapse statement.

5.8 Conditional Complexity Consequence

Corollary 5.2 (Conditional NP-Complete Consequence)**.** *Let $L^* \subseteq \{0,1\}^*$ be an NP-complete language. If $L^* \in \mathbf{P}_{\text{CPR}}$, then $\mathbf{P} = \mathbf{NP}$.*

Proof. By Corollary 5.1, $\mathbf{P}_{\text{CPR}} \subseteq \mathbf{P}$; therefore $L^* \in \mathbf{P}_{\text{CPR}}$ implies $L^* \in \mathbf{P}$. Since L^* is NP-complete, every language in $\mathbf{NP}$ is polynomial-time many-one reducible to L^*, and because $L^* \in \mathbf{P}$, it follows that $\mathbf{NP} \subseteq \mathbf{P}$. The standard inclusion $\mathbf{P} \subseteq \mathbf{NP}$ also holds. Consequently $\mathbf{P} = \mathbf{NP}$. □

Remark 5.1 (Unproved Premise)**.** The manuscript does not establish the hypothesis $L^* \in \mathbf{P}_{\text{CPR}}$ for any NP-complete language L^*; in particular, no NP-complete language is proved in this work to possess a uniform CPR reconstruction model satisfying BC1–BC4 together with uniform polynomial bounds for every component of the complete runtime model. The preceding corollary is therefore a conditional statement only and does not establish $\mathbf{P} = \mathbf{NP}$.

5.9 Conclusion

The four boundary conditions define the structural, constructibility, correctness, and semantic-control requirements of a CPR reconstruction model: BC1, explicit uniform structural representation; BC2, polynomial uniform constructibility; BC3, task-relative soundness and completeness; BC4, polynomial semantic-state and transition-structure control. They do not replace

the complete runtime model; a polynomial-time conclusion additionally requires fixed polynomial functions $p_j : \mathbb{N} \to \mathbb{N}$, $j \in J$, such that, uniformly for every encoded instance P_x, $N_j(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_j(|x|)$ for every $j \in J$. Accordingly, the complete sufficient tractability condition is

$$\Big[\exists \tau \, \exists \mathfrak{M} \, \exists \mathcal{A}_{\text{Preprocess}} \, \exists \{p_j\}_{j \in J} \, \forall x \in \{0,1\}^* : \\ \pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M}) \wedge \text{BC1} \wedge \text{BC2} \wedge \text{BC3} \wedge \text{BC4} \\ \wedge \bigwedge_{j \in J} [N_j(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_j(|x|)] \Big] \Longrightarrow L_{\mathcal{F}} \in \mathbf{P},$$

where $\mathcal{A}_\pi$ is the ordering-supply component of $\mathcal{A}_{\text{Preprocess}}$. The order of quantifiers is essential: the task, reconstruction model, preprocessing algorithm, admissible ordering-supply rule, and six polynomial functions must be fixed for the complete encoded family before the individual instance x is selected. Instance-dependent reconstruction models, ordering rules, or polynomial bounds do not establish uniform polynomial-time tractability.

The class $\mathbf{P}_{\text{CPR}}$ contains exactly the decision languages certified by such a uniform CPR model. Therefore $\mathbf{P}_{\text{CPR}} \subseteq \mathbf{P}$, and for an NP-complete language the corresponding complexity consequence remains strictly conditional, L^* NP-complete $\wedge\, L^* \in \mathbf{P}_{\text{CPR}} \Longrightarrow \mathbf{P} = \mathbf{NP}$, a premise not established in this manuscript; consequently no conclusion $\mathbf{P} = \mathbf{NP}$ is proved. The following chapters examine representative problem classes in order to identify their reconstruction orderings, task-sufficient active interfaces, reachable semantic states, reconstruction-safe signatures, reconstruction operators, and problem-dependent width expressions. Those realizations illustrate how the boundary conditions may be studied explicitly; they do not by themselves establish CPR tractability for the corresponding unrestricted problem classes.

6 Problem-Class Realizations of CPR-Width

The mathematical theory developed in the preceding chapters is intended to apply to declared reconstruction models rather than to one isolated problem domain. This chapter studies representative realizations of Controlled Partial Reconstruction Width for Boolean satisfiability, finite constraint satisfaction, graph coloring, tensor representations, and digital arithmetic. Throughout, let P denote a finite problem instance, let τ be a fixed declared computational task, let $\mathfrak{M}$ be a fixed reconstruction model, and let $\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})$ be an admissible reconstruction ordering; the dependence on τ and $\mathfrak{M}$ may be suppressed.

A central distinction is required throughout. For every problem-class realization, the visibly defined structural boundary is first treated as a candidate interface $\widetilde{B}_i^{\pi,\tau}$, to be distinguished from the cardinality-minimal task-sufficient interface $B_i^{\pi,\tau}$ of Chapter 2. Unless task sufficiency and cardinality minimality have been proved, one may conclude only $b_i^{\pi,\tau}(P) \le |\widetilde{B}_i^{\pi,\tau}|$, and therefore $\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \le \max_i |\widetilde{B}_i^{\pi,\tau}|$; equality requires an additional minimality proof.

6.1 Boolean Satisfiability

Let $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a Boolean formula over $V(\Phi) = \{x_1, \dots, x_n\}$, and let $\pi = (x_{\pi(1)}, \dots, x_{\pi(n)})$ be an admissible variable ordering, with $V_i^\pi = \{x_{\pi(1)}, \dots, x_{\pi(i)}\}$ and $\overline{V_i^\pi} = V(\Phi) \setminus V_i^\pi$. Define the structural SAT candidate interface

$$\widetilde{B}_i^{\Phi,\pi} = \{x \in V_i^\pi : x \text{ occurs in a clause whose residual status is not yet fixed}\},$$

the set of processed variables whose assigned values may still influence residual clauses; it is a natural structural candidate interface, but not automatically the unique or cardinality-minimal task-sufficient one. For a reachable prefix assignment $r \in \mathcal{R}_i^\pi(\Phi)$, define the exact satisfying-extension set $E_i^{\Phi,\pi}(r) = \{s \in \prod_{x \in \overline{V_i^\pi}} \{0,1\} : r \cup s \models \Phi\}$; equality of these sets may be used as one particular reconstruction-safe prefix signature, $\mathrm{Sig}_{\mathrm{SAT},i}^\pi(r) = E_i^{\Phi,\pi}(r)$. This signature is stronger than the mere current decision value $\mathbf{1}[E_i^{\Phi,\pi}(r) \neq \varnothing]$: because it records the complete residual satisfying-extension set, it preserves the decision answer, admissible next values, and successor residual classes, and so satisfies conditions S1–S3 of Chapter 2. If the candidate interface $\widetilde{B}_i^{\Phi,\pi}$ determines this signature, then it is task-sufficient and $b_i^{\pi,\mathrm{SAT}}(\Phi) \le |\widetilde{B}_i^{\Phi,\pi}|$, so $\widehat{\omega}_{\mathrm{rec}}(\Phi, \pi, \mathrm{SAT}; \mathfrak{M}) \le \max_i |\widetilde{B}_i^{\Phi,\pi}|$. An equality claim requires a proof that no smaller processed-variable set determines the same signature; this realization does not claim that unrestricted SAT has bounded CPR-Width or admits polynomial CPR reconstruction.

6.2 Constraint Satisfaction Problems

Let $P_{\mathrm{CSP}} = (V, D, \mathcal{C})$ be a finite CSP instance with $V = \{x_1, \dots, x_n\}$, $D = (D_{x_1}, \dots, D_{x_n})$, and $\mathcal{C} = \{C_1, \dots, C_m\}$; for a constraint C, let $\mathrm{scope}(C) \subseteq V$ denote its variable scope. For an admissible ordering π, define the structural CSP candidate interface

$$\widetilde{B}_i^{P_{\mathrm{CSP}},\pi} = \left\{x \in V_i^\pi : \exists C \in \mathcal{C},\ x \in \mathrm{scope}(C),\ \mathrm{scope}(C) \cap \overline{V_i^\pi} \neq \varnothing\right\},$$

the processed variables participating in constraints whose scopes still contain unresolved variables. For a reachable prefix assignment r, a strong exact residual signature may be defined by $\mathrm{Sig}_{\mathrm{CSP},i}^\pi(r) = \{s \in \prod_{x \in \overline{V_i^\pi}} D_x : r \cup s \text{ satisfies every constraint in } \mathcal{C}\}$; for counting, optimization, or another declared task, the signature may instead preserve the corresponding task value together with the transition information required by S1–S3. If $\widetilde{B}_i^{P_{\mathrm{CSP}},\pi}$ determines the declared safe signature, then $b_i^{\pi,\tau}(P_{\mathrm{CSP}}) \le |\widetilde{B}_i^{P_{\mathrm{CSP}},\pi}|$, hence $\widehat{\omega}_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi, \tau; \mathfrak{M}) \le \max_i |\widetilde{B}_i^{P_{\mathrm{CSP}},\pi}|$. The structural boundary is a certified upper bound once task sufficiency has been established; it becomes the exact CPR interface size only after a cardinality-minimality proof.

6.3 Graph Coloring

Let $G = (V, E)$ be a finite graph and $K = \{1, \dots, k\}$ the available color set. For an ordering $\pi = (v_{\pi(1)}, \dots, v_{\pi(|V|)})$, with $V_i^\pi = \{v_{\pi(1)}, \dots, v_{\pi(i)}\}$ and $\overline{V_i^\pi} = V \setminus V_i^\pi$, define the structural graph-coloring candidate interface

$$\widetilde{B}_i^{G,\pi} = \left\{ v \in V_i^\pi : \exists w \in \overline{V_i^\pi} \text{ with } \{v, w\} \in E \right\},$$

exactly the classical vertex-separation boundary of π. The equality $b_i^{\pi,\tau}(G) = |\widetilde{B}_i^{G,\pi}|$ does not follow solely from this structural observation, since semantic redundancy may permit a smaller task-sufficient coordinate set; the generally valid relation is $b_i^{\pi,\tau}(G) \leq |\widetilde{B}_i^{G,\pi}|$ whenever the structural boundary determines the selected reconstruction-safe signature, and consequently $\widehat{\omega}_{\mathrm{rec}}(G, \pi, \tau; \mathfrak{M}) \leq \max_i |\widetilde{B}_i^{G,\pi}|$. The right-hand side is the ordering-specific vertex-separation value, the left-hand side the minimum task-sufficient CPR interface width; they may coincide for a particular realization, but this equality requires proof. The role of this realization is therefore not to rename vertex separation, but to embed a classical structural boundary into a task-relative reconstruction model with semantic quotienting and complete runtime accounting.

6.3.1 Quantitative Verification for C_4

Consider the four-vertex cycle $C_4 = (V, E)$ with $V = \{v_1, v_2, v_3, v_4\}$ and three available colors. The raw assignment count is $N_{\mathrm{raw}} = 3^4 = 81$. Let $\Phi(C_4) = \{c : V \to \{1, 2, 3\} : c(u) \neq c(v) \text{ for every } \{u, v\} \in E\}$ be the set of proper three-colorings; the number of admissible colorings is $N_{\mathrm{adm}} = |\Phi(C_4)| = 18$. Fix the partial coloring $f(v_1) = 1$, $f(v_2) = 2$; the admissible completions are $(c(v_3), c(v_4)) \in \{(1, 2), (1, 3), (3, 2)\}$, so $N_{\mathrm{rec}}(f) = 3$. The three quantities describe different spaces, $N_{\mathrm{raw}} = 81 \neq N_{\mathrm{adm}} = 18 \neq N_{\mathrm{rec}}(f) = 3$: N_{raw} is the cardinality of the complete raw assignment space, N_{adm} that of the admissible proper-coloring space, and $N_{\mathrm{rec}}(f)$ that of the reconstruction fiber induced by the fixed partial coloring f. These are finite cardinalities, not CPR-Width values, and must not be identified with the active-interface width, the minimum task-sufficient interface size, or the semantic-state count. They distinguish the raw assignment space, the admissible coloring space, and one reconstruction fiber, but are not direct runtime comparisons. The admissible-to-fiber factor is $K_{\mathrm{adm/fiber}} = 18/3 = 6$, and the raw-to-fiber factor is $K_{\mathrm{raw/fiber}} = 81/3 = 27$.

6.4 Common Reconstruction Pattern

The representative problem classes considered in this chapter possess different structural candidate interfaces, semantic interpretations, and instance-specific width expressions. For every declared realization, one must distinguish the structural candidate interface $\widetilde{B}_i^{\pi,\tau}$ from the cardinality-minimal task-sufficient interface $B_i^{\pi,\tau}$. Table 6.1 records the structural realizations and the corresponding instance-specific CPR-Width notations.

6.5 Theory of Controlled and Constant CPR-Width

The width expressions in Table 6.1 are problem-class-specific instantiations of the general CPR-Width definition, not universal numerical constants for the unrestricted problem classes: $\omega_{\mathrm{rec}}(\Phi, \pi)$, $\omega_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi)$, $\omega_{\mathrm{rec}}(G, \pi)$, $\omega_{\mathrm{rec}}(T, \pi)$, and $\omega_{\mathrm{rec}}(A, \pi)$ denote the widths of particular represented instances, declared tasks, reconstruction models, and admissible orderings; they identify which CPR-Width expression must be determined for each problem-class realization.

6.5.1 Theory of a Partially Reconstructible Subclass

Let $\mathcal{C}$ be a combinatorial problem class. A controlled partially reconstructible subclass $\mathcal{C}' \subseteq \mathcal{C}$ may exist when every instance $P \in \mathcal{C}'$ admits an admissible reconstruction ordering $\pi_P \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})$ such that its CPR-Width is controlled by a declared function, $\omega_{\mathrm{rec}}(P, \pi_P, \tau; \mathfrak{M}) \leq w_{\mathcal{C}'}(\ell(P))$, where $\ell(P)$ is the encoding length and $w_{\mathcal{C}'}$ is a declared width-bound function for the subclass. The width condition alone does not establish polynomial reconstructibility:

Table 6.1: Problem-class realizations of Controlled Partial Reconstruction Width.

Problem class	**Active interface**	**Semantic interpretation**	**Instance-specific width**
SAT	Assigned variables still occurring in unresolved clauses.	Future satisfying extensions.	$\omega_{\mathrm{rec}}(\Phi, \pi)$
CSP	Processed variables shared with unresolved constraints.	Future globally compatible assignments.	$\omega_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi)$
Graph coloring	Processed vertices adjacent to unresolved vertices.	Future proper colorings.	$\omega_{\mathrm{rec}}(G, \pi)$
Tensor	Active indices affecting unresolved structural classes.	Future output-safe tensor states.	$\omega_{\mathrm{rec}}(T, \pi)$
Full adder	Task-relevant local state sufficient for exact output reconstruction.	Future output behavior.	$\omega_{\mathrm{rec}}(A, \pi)$

all CPR boundary conditions must additionally hold, $\mathrm{BC}_1 \wedge \mathrm{BC}_2 \wedge \mathrm{BC}_3 \wedge \mathrm{BC}_4$, and all six components of the complete CPR cost model must be polynomially bounded, so that for every $j \in \{\mathrm{Build}, \mathrm{Reduce}, \mathrm{Transition}, \mathrm{Reconstruct}, \mathrm{Verify}, \mathrm{Output}\}$ there exists a polynomial p_j with $N_j(P, \pi_P, \tau; \mathfrak{M}) \leq p_j(\ell(P))$; it then follows that there exists a polynomial p_{CPR} satisfying $N_{\mathrm{CPR}}(P, \pi_P, \tau; \mathfrak{M}) \leq p_{\mathrm{CPR}}(\ell(P))$.

The complete theory may therefore be written compactly as

$$\exists \boldsymbol{\mathcal{C}' \subseteq \mathcal{C}} \quad \forall \boldsymbol{P \in \mathcal{C}'} \quad \exists \boldsymbol{\pi_P \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})} : \quad \boldsymbol{\omega_{\mathrm{rec}}(P, \pi_P, \tau; \mathfrak{M}) \leq w_{\mathcal{C}'}(\ell(P))}$$

together with

$$\mathbf{BC_1 \wedge BC_2 \wedge BC_3 \wedge BC_4} \qquad \text{and} \qquad \boldsymbol{N_{\mathrm{CPR}}(P, \pi_P, \tau; \mathfrak{M}) \in \ell(P)^{O(1)}}.$$

6.5.2 Constant-Width Special Case

A constant-width subclass is a special case of the controlled-width theory. If there exists a constant $k_{\mathcal{C}'} \in \mathbb{N}_0$, independent of the instance size, such that $\omega_{\mathrm{rec}}(P, \pi_P, \tau; \mathfrak{M}) \leq k_{\mathcal{C}'}$ for every $P \in \mathcal{C}'$, then $\mathcal{C}'$ has uniformly bounded or constant CPR-Width under the declared task, model, and ordering rule; equivalently, $w_{\mathcal{C}'}(\ell(P)) = k_{\mathcal{C}'}$. For bounded local domains, $q(P) \leq q_0$, the width-dependent state factor is then bounded by $q(P)^{\omega_{\mathrm{rec}}(P, \pi_P, \tau; \mathfrak{M})+1} \leq q_0^{k_{\mathcal{C}'}+1}$, constant with respect to the encoding length. This does not by itself prove polynomial runtime: the construction of the ordering, interfaces, signatures, quotient states, reconstruction, verification, and output must still satisfy the complete polynomial-cost conditions.

6.5.3 Meaning of the Theory

The theory does not claim that every unrestricted problem class has one fixed constant CPR-Width. It claims that subclasses may exist whose instances admit a suitable admissible reconstruction ordering, have controlled or constant CPR-Width, satisfy all four CPR boundary conditions, and admit complete polynomial CPR reconstruction for the declared task. Thus, the correct statement is not that every problem class has one universal constant width, but rather

a problem class may contain subclasses with controlled or constant CPR-Width.

For Table 6.1, this means that $\omega_{\mathrm{rec}}(\Phi, \pi)$, $\omega_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi)$, $\omega_{\mathrm{rec}}(G, \pi)$, $\omega_{\mathrm{rec}}(T, \pi)$, and $\omega_{\mathrm{rec}}(A, \pi)$ are the corresponding problem-class-specific width expressions; for suitable subclasses, these expressions must be calculated, estimated, or bounded. The table records different realizations of one general CPR-Width theory.

6.6 Common Reconstruction Chain

Despite their structural differences, the representative realizations follow the common pattern

$$\text{Ordering} \longrightarrow \text{Structural Candidate Interface} \longrightarrow \text{Task-Sufficient Interface}$$

followed by

$$\text{Reachable Interface States} \longrightarrow \text{Semantic Quotient} \longrightarrow \text{Reconstruction},$$

or, in symbolic form, $\pi \to \widetilde{B}_i^{\pi,\tau} \to B_i^{\pi,\tau} \to U_i^{\mathrm{reach}} \to S_i \to \mathrm{Output}_\tau$. The candidate interface is problem-class dependent; task sufficiency, minimality, semantic quotienting, and complete runtime accounting are governed by the general CPR-WIDTH theory.

6.7 Tensor Representations

Let $T_P : I(P) \times J(P) \times K(P) \to \mathcal{A}$ be a finite structural tensor associated with a problem instance P; a reconstruction ordering may process tensor indices, fragments, or local structural relations. Define a structural tensor candidate interface

$$\widetilde{B}_i^{T_P,\pi} = \{\alpha \in V_i^\pi : \alpha \text{ still influences an unresolved tensor interaction}\}.$$

The exact interpretation of α depends on the declared tensor representation: it may denote an index, a tensor fragment, a local compatibility relation, or another explicitly encoded reconstruction component. A reconstruction-safe tensor signature must preserve the task-relevant future tensor behavior required by S1–S3, for example all admissible future tensor completions, a residual output-equivalence class, a task-relevant structural output vector, or another certified transition-congruent signature. If the structural tensor boundary determines the selected safe signature, then $b_i^{\pi,\tau}(T_P) \le |\widetilde{B}_i^{T_P,\pi}|$ and $\widehat{\omega}_{\mathrm{rec}}(T_P, \pi, \tau; \mathfrak{M}) \le \max_i |\widetilde{B}_i^{T_P,\pi}|$. Tensor representation supplies a structural carrier; it does not by itself prove a small CPR-Width, efficient semantic quotienting, or polynomial runtime.

6.8 Full Adder

The full adder provides a finite output-preserving quotient. Let $\mathbb{B} = \{0,1\}$ and $X_{\mathrm{FA}} = \mathbb{B}^3$; an input state has the form $x = (a, b, c_{\mathrm{in}})$, and the full-adder output map is $F_{\mathrm{FA}} : \mathbb{B}^3 \to \mathbb{B}^2$. Define the Hamming-weight projection $C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) = a + b + c_{\mathrm{in}}$, so $C_{\mathrm{FA}} : \mathbb{B}^3 \to \{0,1,2,3\}$, and $Q_{\mathrm{FA}}(h) = (h \bmod 2, \lfloor h/2 \rfloor)$. Then $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$: the eight input states collapse into four output-equivalence classes, $8 \to 4$. This is an exact finite output quotient; it should not automatically be identified with $S_{\mathrm{CPR}} = 4$ unless the full adder is embedded into a fully declared staged CPR model. For the isolated finite output task, the exact statement is $S_{\mathrm{out}}(P_{\mathrm{FA}}) = 4$. A later chapter may identify this value with a CPR semantic-state count after specifying the complete reconstruction-stage structure, task, ordering, and model.

6.9 Synthesis, Scope, and Non-Claims

The problem classes considered in this chapter possess different structural boundaries: for SAT, the candidate boundary is induced by residual clauses; for CSP, by unresolved constraint scopes; for graph coloring, by edges crossing the processed and unresolved vertex sets; for tensor representations, by unresolved structural interactions; for digital arithmetic, by the local state required for exact output behavior. These structural boundaries provide candidate interfaces; they become certified CPR interfaces only after task sufficiency has been proved, and exact minimum CPR interfaces only after cardinality minimality has also been proved. Accordingly, every problem-class realization should establish, in order: an explicit component representation; a declared task; an admissible reconstruction ordering; a structural candidate interface; a reconstruction-safe signature satisfying S1–S3; task sufficiency of the candidate interface; cardinality minimality, if exact width equality is claimed; semantic-state and transition bounds; and all six complete runtime-cost bounds.

The realizations in this chapter do not establish that the unrestricted SAT, CSP, graph-coloring, tensor, or arithmetic problem classes possess small CPR-Width; that every structural candidate interface is minimal; that semantic signatures are always computable in polynomial time; that a favorable ordering is always polynomially constructible; or polynomial runtime from width alone. The supported conclusion is narrower:

each problem class admits a problem-specific realization of the general CPR interface and width framework .

A tractability conclusion is permitted only for explicitly defined subclasses satisfying controlled width, BC1–BC4, and all six complete polynomial runtime bounds.

6.10 Conclusion

This chapter developed representative realizations of CPR-Width for SAT, CSP, graph coloring, tensor structures, and the full adder. Table 6.1 records the corresponding problem-class-specific width expressions, $\omega_{\mathrm{rec}}(\Phi, \pi)$, $\omega_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi)$, $\omega_{\mathrm{rec}}(G, \pi)$, $\omega_{\mathrm{rec}}(T, \pi)$, and $\omega_{\mathrm{rec}}(A, \pi)$; these do not denote one fixed universal width for an entire unrestricted problem class, but identify the CPR-Width quantity that must be determined for the declared instance, task, model, and ordering.

A problem class may contain a controlled partially reconstructible subclass $\mathcal{C}' \subseteq \mathcal{C}$ whose instances admit suitable orderings and satisfy $\omega_{\mathrm{rec}}(P, \pi_P, \tau; \mathfrak{M}) \leq w_{\mathcal{C}'}(\ell(P))$; a constant-width subclass is obtained when $w_{\mathcal{C}'}(\ell(P)) = k_{\mathcal{C}'}$ for a fixed constant $k_{\mathcal{C}'}$. Neither controlled width nor constant width alone implies polynomial runtime. The complete sufficient condition remains

$$\mathbf{BC_1} \wedge \mathbf{BC_2} \wedge \mathbf{BC_3} \wedge \mathbf{BC_4} \wedge \boldsymbol{N_{\mathrm{CPR}}(P, \pi_P, \tau; \mathfrak{M}) \in \ell(P)^{O(1)}}.$$

Thus, Chapter 6 identifies the structural realization of CPR-Width across several problem classes and formulates the theory under which controlled or constant-width subclasses may become completely CPR-tractable.

7 Adder and Arithmetic Reconstruction

Digital addition provides a finite and exactly verifiable realization of Controlled Partial Reconstruction. The full-adder and ripple-adder constructions are standard objects in digital arithmetic and combinational circuit design; their role here is not to introduce a new addition algorithm, but to verify that the CPR-WIDTH framework reproduces the known local and linear reconstruction structure of binary addition under an explicit component representation, semantic model, and complete operation-count analysis.

Two formally different constructions are distinguished. First, the isolated full adder admits an exact output-preserving quotient from eight input states to four Hamming-weight classes; these classes are treated as an output quotient and are not identified automatically with a stage-wise CPR semantic-state count. Second, the n-bit ripple-adder family admits a uniform staged transducer model whose future behavior is controlled by one Boolean carry state; the declared task is exact evaluation of the addition map

$$\tau_{\mathrm{add}} = \text{exact computation of } (s_0, s_1, \ldots, s_{n-1}, c_n) \text{ from } (A, B, c_0).$$

All operation-count statements in this chapter use the fixed elementary-operation accounting model of Chapter 4. This chapter develops the arithmetic reconstruction model and proves its structural and asymptotic properties; Chapter 8 records the exhaustive finite verification of the isolated full-adder quotient.

7.1 Full-Adder Map and Hamming-Weight Factorization

Let $\mathbb{B} = \{0, 1\}$. A local full-adder input state is $x = (a, b, c_{\mathrm{in}}) \in \mathbb{B}^3$, where a and b are input bits and c_{in} is the incoming carry bit; the corresponding output state is $F_{\mathrm{FA}}(x) = (s, c_{\mathrm{out}}) \in \mathbb{B}^2$, where s is the sum bit and c_{out} the outgoing carry bit. The full-adder map $F_{\mathrm{FA}} : \mathbb{B}^3 \to \mathbb{B}^2$ is defined by $s = a \oplus b \oplus c_{\mathrm{in}}$ and $c_{\mathrm{out}} = ab \vee ac_{\mathrm{in}} \vee bc_{\mathrm{in}}$, equivalently $a + b + c_{\mathrm{in}} = s + 2c_{\mathrm{out}}$; the complete truth table is shown in Table 7.1.

Table 7.1: Full-adder truth table.

a	b	c_{in}	s	c_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Define the Hamming-weight projection $C_{\mathrm{FA}} : \mathbb{B}^3 \to \{0, 1, 2, 3\}$ by $C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) = a + b + c_{\mathrm{in}}$; the eight Boolean input states are mapped to four Hamming-weight states, $8 \to 4$. Let $Z_{\mathrm{FA}} = \{0, 1, 2, 3\}$ be the reduced Hamming-weight space and define $Q_{\mathrm{FA}} : Z_{\mathrm{FA}} \to \mathbb{B}^2$ by $Q_{\mathrm{FA}}(h) = (h \bmod 2, \lfloor h/2 \rfloor)$; then $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$.

Theorem 7.1 (Exact Full-Adder Factorization)**.** *For every input state $x = (a, b, c_{\mathrm{in}}) \in \mathbb{B}^3$, the complete full-adder output is determined uniquely by the Hamming weight $h = C_{\mathrm{FA}}(x)$; consequently $F_{\mathrm{FA}}(x) = Q_{\mathrm{FA}}(C_{\mathrm{FA}}(x))$.*

Proof. Let $h = a + b + c_{\mathrm{in}}$; since $h \in \{0, 1, 2, 3\}$, its parity determines the sum bit, $s = h \bmod 2$, and its integer quotient by two determines the outgoing carry, $c_{\mathrm{out}} = \lfloor h/2 \rfloor$. Therefore the output pair (s, c_{out}) is uniquely determined by h, so $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$. □

The factorization preserves the complete declared output; it does not preserve the original input state uniquely. For every $h \in Z_{\mathrm{FA}}$, the Hamming-weight fiber is $\Omega_{\mathrm{FA}}(h) = C_{\mathrm{FA}}^{-1}(\{h\})$; the four fibers are $\Omega_{\mathrm{FA}}(0) = \{(0,0,0)\}$, $\Omega_{\mathrm{FA}}(1) = \{(1,0,0),(0,1,0),(0,0,1)\}$, $\Omega_{\mathrm{FA}}(2) = \{(1,1,0),(1,0,1),(0,1,1)\}$, and $\Omega_{\mathrm{FA}}(3) = \{(1,1,1)\}$, with cardinalities $(1,3,3,1)$, so the maximum fiber size is $d_{\mathrm{rec}}^{\max}(P_{\mathrm{FA}}) = 3$. The map Q_{FA} is injective on Z_{FA} ($0 \mapsto (0,0)$, $1 \mapsto (1,0)$, $2 \mapsto (0,1)$, $3 \mapsto (1,1)$), so $C_{\mathrm{FA}}(x) = C_{\mathrm{FA}}(y) \Leftrightarrow F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y)$: the Hamming-weight fibers are exactly the output-equivalence classes of the isolated full adder.

For the isolated full-adder output task, define $x \equiv_{\mathrm{out}} y \Leftrightarrow F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y)$; the output quotient $\mathbb{B}^3/\equiv_{\mathrm{out}} = \{\Omega_{\mathrm{FA}}(0), \ldots, \Omega_{\mathrm{FA}}(3)\}$ has cardinality $S_{\mathrm{out}}(P_{\mathrm{FA}}) = |\mathbb{B}^3/\equiv_{\mathrm{out}}| = 4$, with binary output-state dimension $D_{\mathrm{out}}(P_{\mathrm{FA}}) = \lceil \log_2 S_{\mathrm{out}}(P_{\mathrm{FA}}) \rceil = 2$. The quantities S_{out} and D_{out} refer to the isolated output quotient and are not identified automatically with S_{CPR} and D_{CPR}.

7.2 Ripple-Adder Function Family

For every $n \geq 1$, define the n-bit ripple-adder function $F_{\mathrm{add},n} : \mathbb{B}^n \times \mathbb{B}^n \times \mathbb{B} \to \mathbb{B}^{n+1}$, with input (A, B, c_0), $A = (a_{n-1} \cdots a_1 a_0)_2$, $B = (b_{n-1} \cdots b_1 b_0)_2$, and output $(s_0, s_1, \ldots, s_{n-1}, c_n)$. For every position $i \in \{0, \ldots, n-1\}$, define $h_i = a_i + b_i + c_i$, $s_i = h_i \bmod 2$, and $c_{i+1} = \lfloor h_i/2 \rfloor$; the complete arithmetic identity is $A + B + c_0 = \sum_{i=0}^{n-1} s_i 2^i + c_n 2^n$. The CPR model developed below is a uniform transducer model for the complete function $F_{\mathrm{add},n}$: a fixed encoded input (A, B, c_0) determines one deterministic path through this transducer.

Define the carry-state alphabet $\mathcal{Q}_{\mathrm{carry}} = \{0, 1\}$ and local input alphabet $\mathcal{X}_{\mathrm{local}} = \mathbb{B}^2$, with local input label $\alpha_i = (a_i, b_i) \in \mathcal{X}_{\mathrm{local}}$. Define the carry transition map $\delta_{\mathrm{carry}} : \mathcal{Q}_{\mathrm{carry}} \times \mathcal{X}_{\mathrm{local}} \to \mathcal{Q}_{\mathrm{carry}}$ by $\delta_{\mathrm{carry}}(c, (a, b)) = \lfloor (a + b + c)/2 \rfloor$, and the local output map $\mu_{\mathrm{carry}} : \mathcal{Q}_{\mathrm{carry}} \times \mathcal{X}_{\mathrm{local}} \to \mathbb{B}$ by $\mu_{\mathrm{carry}}(c, (a, b)) = (a + b + c) \bmod 2$; thus every local step is $(c_i, (a_i, b_i)) \mapsto (s_i, c_{i+1})$. The transducer has two carry states, and for every carry state four local input labels are possible, so its complete local labeled transition table contains at most $2 \cdot 4 = 8$ labeled transitions – a property of the uniform family-level transducer, not the number of transitions executed during one fixed addition run.

7.3 CPR Component Representation and Reachability

For the uniform n-bit transducer, define the reconstruction components $\gamma_1, \gamma_2, \ldots, \gamma_n$, where γ_j represents the carry state after processing bit positions $0, \ldots, j-1$; the component set is $V(P_{\mathrm{add},n}) = \{\gamma_1, \ldots, \gamma_n\}$, every component with Boolean domain $D_{\gamma_j} = \mathbb{B}$. The input bits a_i, b_i and the initial carry c_0 are external input labels of a transducer run, not reconstructed component coordinates. The reconstruction ordering is $\pi_{\mathrm{add}} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$, so the number of nonterminal reconstruction stages is $m_{\mathrm{stage}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}) = n$. After stage j, $V_j^{\pi_{\mathrm{add}}} = \{\gamma_1, \ldots, \gamma_j\}$, $0 \leq j \leq n$, with $V_0^{\pi_{\mathrm{add}}} = \varnothing$ and unresolved suffix $\overline{V_j^{\pi_{\mathrm{add}}}} = \{\gamma_{j+1}, \ldots, \gamma_n\}$.

The uniform CPR model ranges over all input words of length n. A prefix carry assignment $r = (\gamma_1, \ldots, \gamma_j)$ is family-reachable when there exist local input labels $(a_0, b_0), \ldots, (a_{j-1}, b_{j-1})$ and an initial carry c_0 that generate the declared carry sequence; the family-level reachable prefix space is denoted $\mathcal{R}_j^{\pi_{\mathrm{add}},\mathrm{unif}}(P_{\mathrm{add},n})$. Both carry values are family-reachable at every nonterminal position, $\{0,1\} \subseteq \{\gamma_j(r) : r \in \mathcal{R}_j^{\pi_{\mathrm{add}},\mathrm{unif}}(P_{\mathrm{add},n})\}$. For one fixed encoded input (A, B, c_0), the deterministic transition rule selects exactly one carry sequence $c_1, c_2, \ldots, c_n$, so the corresponding fixed run contains exactly one reachable prefix state at every stage. The family-level state model and the fixed execution path must not be identified: the former characterizes the uniform CPR interface and semantic alphabet, the latter counts operations for one encoded input.

7.4 Carry Signature, Interface Sufficiency, and Minimality

For every stage $1 \leq j < n$, define the family-level carry signature $\mathrm{Sig}_{\tau_{\mathrm{add}},j}^{\pi_{\mathrm{add}}}(r) = \gamma_j(r)$; at stage zero, the externally supplied initial carry is the current transducer state, and after the terminal carry γ_n has been materialized, the active interface may be released. For every nonterminal

stage $1 \le j < n$, define the candidate interface $\widetilde{B}_j^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}} = \{\gamma_j\}$, and at the initial and terminal stages $\widetilde{B}_0^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}} = \widetilde{B}_n^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}} = \varnothing$.

Proposition 7.1 (Carry-Interface Sufficiency)**.** *For every nonterminal stage $1 \le j < n$, the interface $\widetilde{B}_j^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}} = \{\gamma_j\}$ is task-sufficient for the carry signature.*

Proof. Let r, r' be family-reachable prefix states at stage j satisfying $r|_{\{\gamma_j\}} = r'|_{\{\gamma_j\}}$; then $\gamma_j(r) = \gamma_j(r')$, so $\mathrm{Sig}^{\pi_{\mathrm{add}}}_{\tau_{\mathrm{add}},j}(r) = \gamma_j(r) = \gamma_j(r') = \mathrm{Sig}^{\pi_{\mathrm{add}}}_{\tau_{\mathrm{add}},j}(r')$. Hence the one-component interface is task-sufficient. □

Proposition 7.2 (Carry-Interface Minimality)**.** *For every nonterminal stage $1 \le j < n$, the empty set is not task-sufficient for the uniform ripple-adder transducer. Consequently $b_j^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}}(P_{\mathrm{add},n}) = 1$.*

Proof. By family-level reachability, there exist reachable prefix states r_0, r_1 with $\gamma_j(r_0) = 0$ and $\gamma_j(r_1) = 1$; the two states agree on the empty interface. Choose the next local input label $(a_j, b_j) = (0, 0)$: from carry state 0, the local output is $s_j = 0$ and the successor carry is $\gamma_{j+1} = 0$; from carry state 1, the local output is $s_j = 1$ and the successor carry is again $\gamma_{j+1} = 0$. Thus the two prefix states induce different task-relevant next output behavior, so their reconstruction-safe signatures differ; the empty interface cannot distinguish them and is not task-sufficient. Since the singleton $\{\gamma_j\}$ is task-sufficient, its cardinality is minimal, so $b_j^{\pi_{\mathrm{add}},\tau_{\mathrm{add}}}(P_{\mathrm{add},n}) = 1$. □

It follows that $\widehat{\omega}_{\mathrm{rec}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}, \tau_{\mathrm{add}}; \mathfrak{M}_{\mathrm{add}}) = 1$ and, by the normalization convention, $\omega_{\mathrm{rec}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}, \tau_{\mathrm{add}}; \mathfrak{M}_{\mathrm{add}}) = 0$: normalized width zero still corresponds to one active Boolean carry component.

Proposition 7.3 (Carry Signature Is Reconstruction-Safe)**.** *The carry-signature family satisfies the reconstruction-safety conditions S1–S3 of Chapter 2.*

Proof. Let r, r' be family-reachable prefix states at the same stage j with equal carry signatures, $\gamma_j(r) = \gamma_j(r')$. For every common next input label $(a_j, b_j) \in \mathbb{B}^2$, the local arithmetic value $h_j = a_j + b_j + \gamma_j$ is determined; because the carry values agree, the same input label produces the same sum bit and successor carry, so equal carry signatures preserve the exact residual input-output behavior (S1). The complete local input alphabet $\mathbb{B}^2$ is available from both equal carry states, so the admissible next-label sets coincide (S2). The same next input label applied to equal carry states produces equal successor carry values, so the successor signatures coincide (S3). Hence the carry-signature family is reconstruction-safe. □

7.5 Evaluation, Semantic-State Control, and Runtime

Theorem 7.2 (Ripple-Adder Evaluation)**.** *For every $n \ge 1$ and every fixed encoded input (A, B, c_0), the output $(s_0, s_1, \ldots, s_{n-1}, c_n)$ is determined uniquely by the successive carry transitions.*

Proof. At position $i = 0$, the fixed values a_0, b_0, c_0 determine $h_0 = a_0 + b_0 + c_0$, and hence uniquely $s_0 = h_0 \bmod 2$ and $c_1 = \lfloor h_0/2 \rfloor$. Assume inductively that c_i has been computed correctly; then $h_i = a_i + b_i + c_i$ determines uniquely $s_i = h_i \bmod 2$ and $c_{i+1} = \lfloor h_i/2 \rfloor$. Thus every sum bit and the final carry are determined uniquely. □

The uniform carry-state alphabet contains two states, $S^{\mathrm{unif}}_{\mathrm{carry}} = |\mathcal{Q}_{\mathrm{carry}}| = 2$, a family-level state-alphabet count. For a fixed encoded input, the deterministic evaluation follows one carry state at each stage, so $|S_j^{\mathrm{run}}(A, B, c_0)| = 1$ for every stage of one fixed run, while the uniform family-level bound is $|S_j^{\mathrm{unif}}| \le 2$; consequently $S^{\mathrm{unif}}_{\mathrm{CPR}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}, \tau_{\mathrm{add}}; \mathfrak{M}_{\mathrm{add}}) \le 2$ and $D^{\mathrm{unif}}_{\mathrm{CPR}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}, \tau_{\mathrm{add}}; \mathfrak{M}_{\mathrm{add}}) \le 1$.

Definition 7.1 (Total Stage-Indexed Semantic-State Count)**.** Define $V_{\mathrm{sem}}^{\mathrm{unif}}(P,\pi,\tau;\mathfrak{M}) = \sum_{j=0}^{m_{\mathrm{stage}}(P,\pi)} |S_j^{\mathrm{unif}}(P,\pi,\tau;\mathfrak{M})|$.

For the ripple-adder transducer, $V_{\mathrm{sem}}^{\mathrm{unif}}(P_{\mathrm{add},n},\pi_{\mathrm{add}},\tau_{\mathrm{add}};\mathfrak{M}_{\mathrm{add}}) \le 2(n+1)$.

Definition 7.2 (Total Labeled Semantic-Transition Count)**.** Let $\mathcal{E}_j^{\mathrm{sem}}(P,\pi,\tau;\mathfrak{M})$ denote the set of labeled semantic transitions from stage j to stage $j+1$. Define $E_{\mathrm{sem}}^{\mathrm{unif}}(P,\pi,\tau;\mathfrak{M}) = \sum_{j=0}^{m_{\mathrm{stage}}(P,\pi)-1} |\mathcal{E}_j^{\mathrm{sem}}(P,\pi,\tau;\mathfrak{M})|$.

The uniform ripple-adder transducer has at most eight labeled transition edges per stage, so $E_{\mathrm{sem}}^{\mathrm{unif}}(P_{\mathrm{add},n},\pi_{\mathrm{add}},\tau_{\mathrm{add}};\mathfrak{M}_{\mathrm{add}}) \le 8n$; for a fixed encoded input, exactly one transition edge is executed per bit position, $E_{\mathrm{sem}}^{\mathrm{run}}(A,B,c_0) = n$.

The following bounds concern the evaluation of one fixed encoded input (A,B,c_0) by the uniformly declared ripple-adder model; the runtime analysis does not search over all task-sufficient interfaces, since the one-component carry interface is supplied explicitly and its sufficiency and minimality have already been proved, so no exponential interface-minimality search is included. The representation contains n carry components and n local input pairs; constructing the stage representation and the fixed ordering requires $N_{\mathrm{Build}}(P_{\mathrm{add},n},\pi_{\mathrm{add}},\tau_{\mathrm{add}};\mathfrak{M}_{\mathrm{add}}) = O(n)$, and the fixed interface and signature rules are instantiated once per stage, so $N_{\mathrm{Reduce}} = O(n)$. One deterministic transition is executed per bit position, each using a constant number of Boolean and fixed-width arithmetic operations, so $N_{\mathrm{Transition}} = O(n)$; one sum bit is reconstructed at every bit position, followed by one final carry, so $N_{\mathrm{Reconstruct}} = O(n)$. The identity $a_i + b_i + c_i = s_i + 2c_{i+1}$ can be verified independently at every bit position, so $N_{\mathrm{Verify}} = O(n)$. The explicit output contains $n+1$ bits, $L_{\mathrm{Output}} = n+1$, giving $N_{\mathrm{Output}} = \Theta(n)$ under the elementary output model of Chapter 4. By the complete six-component decomposition, every component is $O(n)$ and the explicit output requires $\Omega(n)$ operations, so $N_{\mathrm{CPR}}(P_{\mathrm{add},n},\pi_{\mathrm{add}},\tau_{\mathrm{add}};\mathfrak{M}_{\mathrm{add}}) = \Theta(n)$. This reproduces the known linear structure of ripple-carry addition inside the complete CPR accounting framework; it is not presented as a new addition algorithm.

7.6 Boundary-Condition Record

The uniformly declared ripple-adder transducer satisfies the four CPR boundary conditions for the exact-output task.

BC1 – Explicit Uniform Structural Representation. The Boolean carry components, their local domains, the reconstruction ordering, the family-level reachability rule, the one-component active interfaces, the carry-signature family, the uniform semantic carry alphabet, the deterministic quotient transition, the local output rule, the verification identity, and the exact output representation are all explicitly defined; therefore BC1 holds.

BC2 – Polynomial Uniform Constructibility. The component representation, ordering, carry interface, transition table, and auxiliary stage data are constructible uniformly in $O(n)$ operations; therefore BC2 holds.

BC3 – Task-Relative Soundness and Completeness. For every bit position, $a_i + b_i + c_i = s_i + 2c_{i+1}$, and the inductive evaluation theorem proves that every fixed input produces exactly one correct output $(s_0,\ldots,s_{n-1},c_n)$; therefore BC3 holds.

BC4 – Polynomial Semantic-State and Transition Control. The uniform family-level semantic quantities satisfy $V_{\mathrm{sem}}^{\mathrm{unif}} \le 2(n+1)$ and $E_{\mathrm{sem}}^{\mathrm{unif}} \le 8n$; for every fixed encoded input, only one state and one transition edge are used at each stage; therefore BC4 holds.

Consequently $N_{\mathrm{CPR}}(P_{\mathrm{add},n},\pi_{\mathrm{add}},\tau_{\mathrm{add}};\mathfrak{M}_{\mathrm{add}}) = \Theta(n)$, so the uniformly encoded ripple-adder family satisfies BC1–BC4 and the complete polynomial-cost requirements for the declared exact-output task.

7.7 Structural Interpretation, Scope, and Conclusion

The isolated full adder demonstrates the exact output-preserving factorization $\mathbb{B}^3 \xrightarrow{C_{\mathrm{FA}}} \{0,1,2,3\} \xrightarrow{Q_{\mathrm{FA}}} \mathbb{B}^2$, with fiber cardinalities $(1,3,3,1)$ and output-quotient count $S_{\mathrm{out}}(P_{\mathrm{FA}}) = 4$. The ripple-adder family demonstrates uniform staged reconstruction through the two-state carry transducer $(c_i, (a_i, b_i)) \mapsto (s_i, c_{i+1})$: at every nonterminal stage, the active task-sufficient component is the current Boolean carry component γ_j, so $\widehat{\omega}_{\mathrm{rec}} = 1$ and $\omega_{\mathrm{rec}} = 0$. The family-level model contains two possible carry states; a fixed encoded input follows one deterministic path through that two-state model.

The arithmetic results establish exact output reconstruction for the isolated full-adder quotient and the uniformly encoded ripple-adder family defined in this chapter. They do not establish that every Boolean circuit admits a Hamming-weight factorization, that every arithmetic circuit possesses constant CPR-Width, that every circuit topology admits a one-component interface, that the finite ratio 8/4 is a measured hardware speedup, that the family-level carry alphabet is the same object as the state set visited during one fixed run, that a local quotient automatically extends to arbitrary circuit networks, or that structural compression alone guarantees a wall-clock advantage. The supported conclusions are limited to the exact isolated full-adder output quotient, the declared uniform ripple-adder transducer, the one-component carry interface, the exact normalized width, the two-state uniform carry alphabet, the linear family-level semantic bounds, and the linear complete operation count for one encoded input.

The uniform semantic representation satisfies $S_{\mathrm{CPR}}^{\mathrm{unif}} \leq 2$, $V_{\mathrm{sem}}^{\mathrm{unif}} \leq 2(n+1)$, $E_{\mathrm{sem}}^{\mathrm{unif}} \leq 8n$; all six runtime components are linearly bounded, and the explicit output requires linear work, so $N_{\mathrm{CPR}}(P_{\mathrm{add},n}, \pi_{\mathrm{add}}, \tau_{\mathrm{add}}; \mathfrak{M}_{\mathrm{add}}) = \Theta(n)$. These results verify the CPR accounting framework on a standard linear arithmetic process; they do not imply corresponding width or runtime properties for arbitrary Boolean or arithmetic circuits. The following chapter records the exhaustive finite verification of the isolated full-adder factorization and its output-equivalence fibers.

8 Finite Verification Records

The preceding chapter established the full-adder factorization symbolically and developed the ripple-adder family as a uniform arithmetic reconstruction model. The purpose of the present chapter is narrower: it verifies the isolated full-adder Hamming-weight factorization by exhaustive enumeration of the complete finite input space. Every full-adder input state, Hamming-weight value, output state, and projection fiber is listed or determined explicitly. Accordingly, this chapter verifies the complete full-adder truth table, the Hamming-weight projection, the projection fibers, the exact output-preserving factorization, and the finite output-equivalence quotient. It does not construct a stage-wise CPR model for the isolated full-adder block: in particular, it does not define a reconstruction-component ordering, a sequence of reachable prefix-state spaces, a sequence of minimum task-sufficient interfaces, or stage-dependent CPR semantic quotient spaces. The verified finite relation is $\mathbb{B}^3 \xrightarrow{C_{\mathrm{FA}}} Z_{\mathrm{FA}} \xrightarrow{Q_{\mathrm{FA}}} \mathbb{B}^2$, where C_{FA} forms Hamming-weight classes and Q_{FA} reconstructs the exact full-adder output pair.

8.1 Purpose and Finite Input/Output Spaces

Finite verification serves two purposes: first, it checks the symbolic full-adder factorization of Chapter 7 against every element of the complete finite input space; second, it distinguishes between complete input states, reduced Hamming-weight states, projection fibers, output-equivalence classes, and full-adder output states. The verification remains finite throughout; no asymptotic tractability conclusion, complete operation-count advantage, measured runtime advantage, or general CPR-Width conclusion is derived from the calculations in this chapter.

Let $\mathbb{B} = \{0, 1\}$; the complete local input space is $X_{\mathrm{FA}} = \mathbb{B}^3$, so $|X_{\mathrm{FA}}| = 2^3 = 8$. Every input state has the form $x = (a, b, c_{\mathrm{in}})$, $a, b, c_{\mathrm{in}} \in \mathbb{B}$, and the full-adder output map is $F_{\mathrm{FA}} : X_{\mathrm{FA}} \to \mathbb{B}^2$, with $F_{\mathrm{FA}}(x) = (s, c_{\mathrm{out}})$, where s is the sum bit and c_{out} the outgoing carry bit.

8.2 Complete Finite Verification Table

Define the Hamming weight of an input state by $h = a + b + c_{\mathrm{in}}$. The complete finite input space and the corresponding outputs are shown in Table 8.1.

Table 8.1: Complete finite verification table for the full-adder Hamming-weight factorization.

a	b	c_{in}	h	s	c_{out}	$Q_{\mathrm{FA}}(h)$
0	0	0	0	0	0	$(0, 0)$
0	0	1	1	1	0	$(1, 0)$
0	1	0	1	1	0	$(1, 0)$
0	1	1	2	0	1	$(0, 1)$
1	0	0	1	1	0	$(1, 0)$
1	0	1	2	0	1	$(0, 1)$
1	1	0	2	0	1	$(0, 1)$
1	1	1	3	1	1	$(1, 1)$

For every row, the arithmetic identity $a + b + c_{\mathrm{in}} = s + 2c_{\mathrm{out}}$ holds, and the last three columns also show directly that $F_{\mathrm{FA}}(x) = Q_{\mathrm{FA}}(h)$ for every one of the eight possible input states.

8.3 Hamming-Weight Projection and Fibers

Define the Hamming-weight projection $C_{\mathrm{FA}} : \mathbb{B}^3 \to \{0, 1, 2, 3\}$ by $C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) = a + b + c_{\mathrm{in}}$, so the reduced state space $Z_{\mathrm{FA}} = \{0, 1, 2, 3\}$ has $|Z_{\mathrm{FA}}| = 4$, the finite projection cardinality pattern $8 \to 4$. Define the output-reconstruction map $Q_{\mathrm{FA}} : Z_{\mathrm{FA}} \to \mathbb{B}^2$ by $Q_{\mathrm{FA}}(h) = (h \bmod 2, \lfloor h/2 \rfloor)$;

the row-wise verification of Table 8.1 yields $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$. The map C_{FA} is not injective, so the original input state cannot generally be reconstructed uniquely from its Hamming weight; the complete output pair, however, is reconstructed uniquely by Q_{FA}.

For every $h \in Z_{\mathrm{FA}}$, define the projection fiber $\Omega_{\mathrm{FA}}(h) = C_{\mathrm{FA}}^{-1}(\{h\})$, which contains every complete input state mapped to the same reduced Hamming-weight state (it does not define a unique inverse of C_{FA}). The four fibers are $\Omega_{\mathrm{FA}}(0) = \{(0,0,0)\}$, $\Omega_{\mathrm{FA}}(1) = \{(1,0,0),(0,1,0),(0,0,1)\}$, $\Omega_{\mathrm{FA}}(2) = \{(1,1,0),(1,0,1),(0,1,1)\}$, and $\Omega_{\mathrm{FA}}(3) = \{(1,1,1)\}$, with cardinalities $(1,3,3,1)$. The fibers form a partition of the complete input space, $\mathbb{B}^3 = \bigsqcup_{h \in Z_{\mathrm{FA}}} \Omega_{\mathrm{FA}}(h)$, and indeed $\sum_{h \in Z_{\mathrm{FA}}} |\Omega_{\mathrm{FA}}(h)| = 1+3+3+1 = 8$; the maximum projection-fiber size is $\max_{h \in Z_{\mathrm{FA}}} |\Omega_{\mathrm{FA}}(h)| = 3$. This is a finite fiber-cardinality statement, not a CPR-Width value, a stage-wise semantic-state count, or a runtime measure.

8.4 Exact Output and Output-Equivalence Verification

Theorem 8.1 (Complete Finite Full-Adder Factorization Verification)**.** *For every input state $x \in \mathbb{B}^3$, the reduced Hamming-weight state $h = C_{\mathrm{FA}}(x)$ uniquely determines the complete full-adder output $F_{\mathrm{FA}}(x) = (s, c_{\mathrm{out}})$. Consequently $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$ on the complete finite input space.*

Proof. Let $h = a + b + c_{\mathrm{in}}$. Since $h \in \{0,1,2,3\}$, its parity determines the sum bit, $s = h \bmod 2$, and its integer quotient by two determines the outgoing carry, $c_{\mathrm{out}} = \lfloor h/2 \rfloor$. Thus $F_{\mathrm{FA}}(x) = (h \bmod 2, \lfloor h/2 \rfloor) = Q_{\mathrm{FA}}(h) = Q_{\mathrm{FA}}(C_{\mathrm{FA}}(x))$. Table 8.1 lists all eight elements of $\mathbb{B}^3$ and confirms the equality for every row; therefore the factorization holds on the complete finite input space. □

For the isolated full-adder output task, define $x \equiv_{\mathrm{out}} y \Leftrightarrow F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y)$; this relation compares complete input states according to their output behavior and is not the general stage-wise CPR semantic equivalence of Chapter 3.

Lemma 8.1 (Equality of Output and Hamming-Weight Classes)**.** *For all $x, y \in \mathbb{B}^3$, $F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y) \Leftrightarrow C_{\mathrm{FA}}(x) = C_{\mathrm{FA}}(y)$.*

Proof. If $C_{\mathrm{FA}}(x) = C_{\mathrm{FA}}(y)$, the factorization $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$ gives $F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y)$. Conversely, the values of Q_{FA} are $Q_{\mathrm{FA}}(0) = (0,0)$, $Q_{\mathrm{FA}}(1) = (1,0)$, $Q_{\mathrm{FA}}(2) = (0,1)$, $Q_{\mathrm{FA}}(3) = (1,1)$, which are pairwise distinct, so $Q_{\mathrm{FA}} : Z_{\mathrm{FA}} \to \mathbb{B}^2$ is bijective and in particular injective; hence $F_{\mathrm{FA}}(x) = F_{\mathrm{FA}}(y)$ implies $C_{\mathrm{FA}}(x) = C_{\mathrm{FA}}(y)$. □

The output quotient $S_{\mathrm{FA}}^{\mathrm{out}} = \mathbb{B}^3/\equiv_{\mathrm{out}}$ therefore has classes precisely the Hamming-weight fibers, $S_{\mathrm{FA}}^{\mathrm{out}} = \{\Omega_{\mathrm{FA}}(0), \ldots, \Omega_{\mathrm{FA}}(3)\}$, so $|S_{\mathrm{FA}}^{\mathrm{out}}| = 4$. The binary output-quotient dimension is $D_{\mathrm{FA}}^{\mathrm{out}} = \lceil \log_2 \max\{1, |S_{\mathrm{FA}}^{\mathrm{out}}|\} \rceil = \lceil \log_2 4 \rceil = 2$; $|S_{\mathrm{FA}}^{\mathrm{out}}| = 4$ counts the output-equivalence classes, and $D_{\mathrm{FA}}^{\mathrm{out}} = 2$ counts the bits required to identify one of those four classes. Neither quantity is asserted here to equal a stage-wise quantity $S_{\mathrm{CPR}}(P_{\mathrm{FA}}, \pi, \tau; \mathfrak{M})$ or $D_{\mathrm{CPR}}(P_{\mathrm{FA}}, \pi, \tau; \mathfrak{M})$, because no staged CPR reconstruction model for the isolated full-adder block is constructed in this chapter.

8.5 Verification Record and Evidence Status

The exhaustive finite verification establishes: the complete input space contains exactly eight states; the Hamming-weight projection produces four reduced states; the projection fibers have cardinalities $(1,3,3,1)$; the four fibers are pairwise disjoint and cover $\mathbb{B}^3$; the maximum projection-fiber size is 3; the full-adder output map factors through the Hamming weight; Q_{FA} is bijective on Z_{FA}; two input states have the same output exactly when they have the same Hamming weight; and the output quotient contains exactly four classes. This is a finite verification record: it establishes the stated properties for the complete isolated full-adder input space, but does not establish an asymptotic theorem for an unrestricted circuit family.

Under the evidence hierarchy introduced in Chapter 9, the present full-adder record has finite-verification status, denoted Evidence Level E1.

Evidence Status: E1 — Finite Verification Evidence

Evidence Level E1 means that a claim is checked on a finite, explicitly stated example. The present record qualifies because all eight input states are enumerated, all four Hamming-weight values are listed, every projection fiber is computed exactly, the factorization is checked on the complete finite input space, and the output-equivalence quotient is determined exactly. The evidence level supports only the corresponding finite conclusions; in particular, E1 $\not\Longrightarrow$ E4, where E4 denotes an asymptotic theorem under explicit assumptions. Likewise, $8 \to 4$ is a finite state-cardinality relation and does not by itself imply an operation-count advantage or a measured wall-clock advantage.

8.6 Scope, Non-Claims, and Conclusion

This chapter verifies only the isolated full-adder output quotient. It does not establish a stage-wise CPR-Width for the isolated full adder, a uniform width theorem for arbitrary circuits, a polynomial-time result for a new problem class, an operation-count speedup relative to a declared baseline, or a measured hardware acceleration. The supported conclusion is exact finite verification of the declared eight-state full-adder model.

The complete isolated full-adder input space has been verified exhaustively. The Hamming-weight projection is $\mathbb{B}^3 \xrightarrow{C_{\mathrm{FA}}} \{0,1,2,3\}$, with fiber-cardinality vector $(1,3,3,1)$, output-reconstruction map $Q_{\mathrm{FA}}(h) = (h \bmod 2, \lfloor h/2 \rfloor)$, and exact factorization $F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$. Every input state in the same projection fiber induces the same output pair; conversely, because Q_{FA} is injective, two input states with the same output belong to the same Hamming-weight fiber, so $\mathbb{B}^3/{\equiv_{\mathrm{out}}} = \{\Omega_{\mathrm{FA}}(0), \ldots, \Omega_{\mathrm{FA}}(3)\}$.

This constitutes exact finite verification of the Hamming-weight factorization, the projection fibers, the output-preservation property, and the output-equivalence quotient. It does not constitute a stage-wise CPR-Width verification, nor does it establish an asymptotic complexity result, a complete operation-count advantage, or a measured wall-clock improvement. The following chapter develops the complexity-theoretic scope, open problems, and future research program of CPR-WIDTH.

9 Complexity-Theoretic Scope, Open Problems, and Future Research

The preceding chapters established Controlled Partial Reconstruction Width as a reconstruction-oriented structural framework for finite combinatorial problems. Throughout this chapter, let P be a finite problem instance, τ the declared computational task, $\mathfrak{M}$ the fixed CPR reconstruction model, and $\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M})$ an admissible reconstruction ordering. The central structural-runtime chain of the manuscript is

$$\boldsymbol{\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \longrightarrow S_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) \longrightarrow N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})} :$$

the first quantity is the normalized CPR-Width, the second the maximum number of semantic quotient states occurring at one reconstruction stage, and the third the complete CPR operation count; when unambiguous, dependence on τ and $\mathfrak{M}$ may be suppressed. This chain expresses structural and accounting dependencies; it does not state that small reconstruction width alone implies polynomial-time tractability. As established in Chapter 4, the complete CPR operation count is $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$; a favorable value of ω_{rec} controls only one structural contribution to the complete analysis, and a polynomial-time conclusion requires uniform polynomial control of every term. Let $\mathcal{J} = \{\mathrm{Build}, \mathrm{Reduce}, \mathrm{Transition}, \mathrm{Reconstruct}, \mathrm{Verify}, \mathrm{Output}\}$ be the runtime-index set.

Let $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$ be a uniformly encoded family of finite decision-problem instances with associated decision language $L_{\mathcal{F}} = \{x \in \{0,1\}^* : P_x \text{ is a yes-instance}\}$. A complete uniform CPR decision model requires one fixed decision task τ, one fixed uniform reconstruction model $\mathfrak{M}$, one deterministic preprocessing algorithm $A_{\mathrm{Preprocess}}$, one deterministic ordering algorithm A_π, one uniform correctness specification, and six fixed polynomial functions $\{p_j\}_{j \in \mathcal{J}}$. For every encoding x, let $\mathcal{D}_x = A_{\mathrm{Preprocess}}(x)$ denote the complete uniformly constructed CPR representation of P_x and $\pi_x = A_\pi(x)$ the produced ordering; the operation counts of $A_{\mathrm{Preprocess}}$ and A_π are included in $N_{\mathrm{Build}}(P_x, \pi_x, \tau; \mathfrak{M})$. The complete uniform tractability requirement is

$$\begin{aligned}
\Big[&\exists \tau \, \exists \mathfrak{M} \, \exists A_{\mathrm{Preprocess}} \, \exists A_\pi \, \exists \{p_j\}_{j \in \mathcal{J}} \, \forall x \in \{0,1\}^* : \; \mathcal{D}_x = A_{\mathrm{Preprocess}}(x) \\
&\wedge \pi_x = A_\pi(x) \in \Pi_{\mathrm{adm}}(P_x, \tau; \mathfrak{M}) \wedge \bigwedge_{k=1}^{4} \mathrm{BC}_k(P_x, \pi_x, \tau; \mathfrak{M}) \\
&\wedge \bigwedge_{j \in \mathcal{J}} \left[N_j(P_x, \pi_x, \tau; \mathfrak{M}) \le p_j(|x|) \right] \Big] \Longrightarrow L_{\mathcal{F}} \in \mathbf{P},
\end{aligned}$$

with the order of quantifiers essential: task, model, preprocessing algorithm, ordering algorithm, correctness specification, and polynomial bounds must be fixed for the complete encoded family before an individual encoding is selected. Under the complete definition of $\mathbf{P}_{\mathrm{CPR}}$ of Chapter 5, which includes BC1–BC4, uniform ordering supply, and polynomial bounds for all six runtime components, $\mathbf{P}_{\mathrm{CPR}} \subseteq \mathbf{P}$; this inclusion is not a complexity-class collapse statement. For an NP-complete language L^*, the implication $L^* \in \mathbf{P}_{\mathrm{CPR}} \Rightarrow \mathbf{P} = \mathbf{NP}$ remains strictly conditional, and no NP-complete language is proved in this work to belong to $\mathbf{P}_{\mathrm{CPR}}$.

Chapter 6 presents structural realization templates, controlled-width subclass theory, and finite illustrative calculations. Chapter 7 establishes symbolic arithmetic reconstruction results and a uniform carry-transducer analysis. Chapter 8 provides an exhaustive finite verification record for the isolated full-adder Hamming-weight factorization. These finite and structural results do not by themselves establish asymptotic tractability for unrestricted problem classes. Count-level gain and measured wall-clock gain must also remain separate: a favorable elementary-operation count is a statement within the declared accounting model, while a measured runtime claim requires a separate implementation, hardware description, software environment, benchmark protocol, and reproducible measurement procedure. The remaining sections formulate the principal mathematical, algorithmic, and experimental research problems of CPR-WIDTH.

9.1 Structural and Algorithmic Open Problems

The first group of open problems concerns the internal mathematical structure of CPR reconstruction models. The central questions are: whether favorable reconstruction orderings can be constructed efficiently; whether exact semantic-state encodings can be computed without exhaustive future enumeration; how tight the width-based state bound is; how CPR-Width relates to classical width parameters; and which nontrivial uniformly encoded families satisfy the complete CPR tractability requirements.

9.1.1 Optimal Reconstruction Orderings

Let $\Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})$ denote the set of admissible reconstruction orderings, assumed nonempty. The minimum normalized CPR-Width is $\omega_{\mathrm{rec}}(P,\tau;\mathfrak{M}) = \min_{\pi\in\Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})} \omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})$, with corresponding unnormalized minimum $\widehat{\omega}_{\mathrm{rec}}(P,\tau;\mathfrak{M}) = \min_{\pi} \widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})$. The definition does not imply that a minimizing ordering can be found efficiently; the exact ordering problem is to determine $\pi^* \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})$ achieving this minimum. A complete algorithmic analysis should distinguish: whether the exact minimum width can be computed in polynomial time for a specified uniformly encoded family; whether one can determine whether $\omega_{\mathrm{rec}} \le k$ for a declared integer k; whether a polynomial-time approximation algorithm exists when exact minimization is hard; whether fixed-parameter algorithms are possible with respect to $\widehat{\omega}_{\mathrm{rec}}$, $q(P)$, or another declared structural parameter; whether useful orderings can be constructed without first constructing the complete reachable prefix-state space or semantic quotient; and whether structural lower bounds on CPR-Width can be obtained directly from the encoded instance.

9.1.2 Width Ratio

Let $\widehat{\pi} \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})$ be a declared admissible ordering. Define the unnormalized width ratio $\alpha_{\mathrm{width}}(P,\widehat{\pi},\tau;\mathfrak{M}) = (\widehat{\omega}_{\mathrm{rec}}(P,\widehat{\pi},\tau;\mathfrak{M})+1)/(\widehat{\omega}_{\mathrm{rec}}(P,\tau;\mathfrak{M})+1)$; adding one guarantees a positive denominator while preserving the exact optimality criterion. This is a structural width ratio, not an operation-count ratio or a measured runtime overhead.

Proposition 9.1 (Width-Ratio Bound)**.** *For every admissible ordering $\widehat{\pi}$, $\alpha_{\mathrm{width}}(P,\widehat{\pi},\tau;\mathfrak{M}) \ge 1$, with equality if and only if $\widehat{\omega}_{\mathrm{rec}}(P,\widehat{\pi},\tau;\mathfrak{M}) = \widehat{\omega}_{\mathrm{rec}}(P,\tau;\mathfrak{M})$.*

Proof. By definition of the minimum unnormalized width, $\widehat{\omega}_{\mathrm{rec}}(P,\widehat{\pi},\tau;\mathfrak{M}) \ge \widehat{\omega}_{\mathrm{rec}}(P,\tau;\mathfrak{M})$; adding one to both sides and dividing by the strictly positive denominator gives $\alpha_{\mathrm{width}}(P,\widehat{\pi},\tau;\mathfrak{M}) \ge 1$, with equality exactly when the two unnormalized width values are equal. □

9.1.3 Efficient Semantic-State Construction

At stage i, the semantic quotient is $S_i(P,\pi,\tau;\mathfrak{M}) = U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})/\equiv^{\tau,\mathfrak{M}}_{P,\pi,i}$, with primary equivalence $u \equiv^{\tau,\mathfrak{M}}_{P,\pi,i} v \Leftrightarrow \widehat{\mathrm{Sig}}^{\pi}_{\tau,i}(u) = \widehat{\mathrm{Sig}}^{\pi}_{\tau,i}(v)$, where $\widehat{\mathrm{Sig}}^{\pi}_{\tau,i} : U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \to \Sigma^{\pi}_{\tau,i}$ is the induced reconstruction-safe interface-signature map: the equivalence relation is defined by reconstruction-safe signatures, not generally by literal equality of complete continuation sets. Exact continuation-set equality may be used as a stronger special signature realization only when it is well defined independently of the selected prefix representative, preserves S1–S3, and the declared task requires that stronger information.

The principal algorithmic question is whether semantic classes can be represented by a finite efficiently computable canonical encoding. Let $\chi_i : U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \to \Gamma_i(P,\pi,\tau;\mathfrak{M})$ be a finite encoding map (the symbol χ_i avoids conflict with the stage-wise branching number λ_i of Chapter 4 and the admissible next-value sets $\Lambda_i^{\pi}(r)$). The encoding is exact if $\chi_i(u) = \chi_i(v) \Leftrightarrow u \equiv^{\tau,\mathfrak{M}}_{P,\pi,i} v$; under this condition, the fibers of χ_i coincide exactly with the semantic equivalence classes, giving a natural bijection $S_i(P,\pi,\tau;\mathfrak{M}) \cong \chi_i(U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}))$. The construction of polynomial-time exact encodings for nontrivial uniformly encoded families remains an open problem.

A one-sided condition must be interpreted carefully. If $\chi_i(u) = \chi_i(v) \Rightarrow u \equiv^{\tau,\mathfrak{M}}_{P,\pi,i} v$, the encoding may refine the semantic quotient by assigning different labels to semantically equiv-

alent states; this may reduce compression but does not merge semantically different states, so $|\chi_i(U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}))| \geq |S_i(P,\pi,\tau;\mathfrak{M})|$. If this implication fails, the encoding may merge states whose reconstruction-safe future behavior differs, which can destroy soundness, completeness, admissible-next-value preservation, or successor congruence. Approximate state merging must therefore be separated from exact CPR semantic quotienting and requires an explicit approximation guarantee.

9.1.4 Width-Bound Slack

The sharp operational Width–State Bound is $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$. Since $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \geq 1$, define the sharp width-bound slack factor $\kappa_{\mathrm{bound}}(P,\pi,\tau;\mathfrak{M}) = q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}/S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})$.

Proposition 9.2 (Width-Bound Slack)**.** *For every finite problem instance, declared task, reconstruction model, and admissible ordering,* $\kappa_{\mathrm{bound}}(P,\pi,\tau;\mathfrak{M}) \geq 1$.

Proof. By the Width–State Bound, $S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M}) \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}$; dividing the upper bound by the positive semantic-state count gives $\kappa_{\mathrm{bound}}(P,\pi,\tau;\mathfrak{M}) \geq 1$. □

A value close to one indicates that the general raw-interface-state bound is close to the observed maximum semantic-state count; a large value indicates that reachability restrictions or semantic quotienting produce a much smaller state space than the raw width bound. The slack factor is not a measured compression speedup, operation-count gain, or wall-clock gain.

9.1.5 Relationship to Classical Width Parameters

CPR-Width is not defined as treewidth, pathwidth, branchwidth, clique-width, or hypertree width. Nevertheless, specific relationships may exist for explicitly declared problem representations and reconstruction tasks. Let $G_P = \mathcal{G}(P)$ be the explicitly declared graph representation associated with P (a primal graph, incidence graph, constraint graph, circuit graph, or another specified construction, depending on the problem class), and let $\mathrm{tw}(G_P)$, $\mathrm{pw}(G_P)$, $\mathrm{bw}(G_P)$, $\mathrm{cw}(G_P)$ denote treewidth, pathwidth, branchwidth, and clique-width. Future research should investigate whether inequalities of the form $\omega_{\mathrm{rec}}(P,\tau;\mathfrak{M}) \leq f(\mathrm{tw}(G_P))$ or $\mathrm{tw}(G_P) \leq g(\omega_{\mathrm{rec}}(P,\tau;\mathfrak{M}))$ hold for explicitly defined functions $f, g : \mathbb{N}_0 \to \mathbb{N}_0$; similar questions may be posed for pathwidth, branchwidth, clique-width, hypertree width, trellis width, contraction width, and other problem-class-specific width notions. No universal comparison is claimed: any theorem relating CPR-Width to a classical width parameter must fix the problem representation, the declared task, the reconstruction model, the admissible-ordering family, the active-interface rule, the reconstruction-safe signature family, and the graph or relational representation $\mathcal{G}(P)$, since the same underlying graph may induce different CPR-Width values for decision, counting, enumeration, optimization, or exact reconstruction tasks. This open problem, together with the informal discussion of treewidth and pathwidth in Chapter 1, is the precise sense in which a formal structural comparison between CPR-Width and classical width parameters is left for future work rather than established in this manuscript.

9.1.6 Identification of Additional CPR-Tractable Families

The present work establishes an exact finite output-factorization result for the isolated full adder and a uniform carry-interface reconstruction result for the ripple-adder family. The broader classification problem is to identify nontrivial uniformly encoded infinite families satisfying BC1–BC4 together with the complete six-component polynomial-cost requirements. For a candidate uniformly encoded family $\mathcal{F} = \{P_x : x \in \{0,1\}^*\}$, a complete CPR-tractability proof must provide one uniform algorithm producing $\pi_x \in \Pi_{\mathrm{adm}}(P_x,\tau;\mathfrak{M})$ and fixed polynomials $p_j : \mathbb{N} \to \mathbb{N}$, $j \in \mathcal{J}$, such that $N_j(P_x,\pi_x,\tau;\mathfrak{M}) \leq p_j(|x|)$ for every encoding x and every j. Promising research areas include restricted SAT and 3-SAT families, bounded-domain CSP families, graph-coloring subclasses, Hamiltonian-path and Hamiltonian-cycle subclasses,

bounded-interface routing and scheduling systems, arithmetic and Boolean circuit families, tensor-network families, database-join families, coding and decoding systems, and other families admitting exact local continuation summaries. No family should be classified as CPR-tractable merely because a compact representation or small structural candidate interface has been identified; the complete requirement remains the uniform tractability condition above.

9.2 Validation, Comparative Analysis, and Future Research Program

The second group of open problems concerns implementation, measurement, comparison, reproducibility, and evidence classification. The objective is not merely to produce a working prototype; a valid prototype should expose every structural and computational layer of the declared CPR process.

9.2.1 Prototype Architecture and Reference Pipeline

For a finite problem instance P, task τ, reconstruction model $\mathfrak{M}$, and ordering π, define the core structural and operation-count record

$$\mathcal{R}_{\mathrm{core}}(P,\pi,\tau;\mathfrak{M}) = \big(\widehat{\omega}_{\mathrm{rec}}, \omega_{\mathrm{rec}}, S_{\mathrm{CPR}}, V_{\mathrm{sem}}, E_{\mathrm{sem}}, m_{\mathrm{stage}},$$
$$N_{\mathrm{Build}}, N_{\mathrm{Reduce}}, N_{\mathrm{Transition}}, N_{\mathrm{Reconstruct}}, N_{\mathrm{Verify}}, N_{\mathrm{Output}}\big),$$

where $V_{\mathrm{sem}}(P,\pi,\tau;\mathfrak{M}) = \sum_{i=0}^{m_{\mathrm{stage}}(P,\pi)} |S_i(P,\pi,\tau;\mathfrak{M})|$ is the total stage-indexed semantic-state count and $E_{\mathrm{sem}}(P,\pi,\tau;\mathfrak{M}) = \sum_{i=0}^{m_{\mathrm{stage}}(P,\pi)-1} |\mathcal{E}_i^{\mathrm{sem}}(P,\pi,\tau;\mathfrak{M})|$ the total semantic-transition count, $\mathcal{E}_i^{\mathrm{sem}}$ denoting the reachable labeled semantic quotient transitions from stage i to stage $i+1$ (interpreted as 0 when $m_{\mathrm{stage}}(P,\pi) = 0$); the superscript "unif" of Chapter 7 denotes their family-level specialization for the uniform ripple-adder transducer. A complete measured implementation record should additionally include

$$\mathcal{R}_{\mathrm{measured}} = (T_{\mathrm{CPR}}, M_{\mathrm{peak}}, L_{\mathrm{Output}}, I_{\mathrm{CPR}}, H, S_{\mathrm{env}}, \mathcal{P}_{\mathrm{measure}}),$$

the measured wall-clock time, peak memory use, encoded output length, implementation identifier, hardware identifier, software-environment identifier, and measurement-protocol identifier, all referring to the same representation, task, model, ordering, correctness requirement, and output requirement. A prototype that reports only the reduced-state count or only the transition count does not test the complete CPR runtime framework.

A reproducible implementation should respect the non-circular construction order of Chapters 2 and 3. The reference execution pipeline is: parse and validate the encoded instance; construct the complete initial CPR representation; construct or read an admissible reconstruction ordering; generate the processed-prefix and unresolved-suffix sequence; generate the reachable prefix-state spaces under the declared reachability and transition rules; compute, retrieve, or certify the reconstruction-safe prefix signatures; determine the family of task-sufficient interfaces at every stage; select a cardinality-minimal task-sufficient interface whenever the exact CPR-Width is required; construct the raw interface-state spaces; project reachable prefix states onto the selected interfaces; construct the induced interface-signature maps; form the semantic quotients induced by the reconstruction-safe signatures (not generally a future-extension-set quotient); construct and process the semantic quotient transitions; perform task-relevant reconstruction; verify soundness and completeness for the declared task; materialize the required output; and record all structural quantities, operation counts, memory use, and measured runtime data. For every stage i, define the stage record $\mathcal{A}_i(P,\pi,\tau;\mathfrak{M}) = (b_i^{\pi,\tau}(P), |U_i(P,\pi,\tau;\mathfrak{M})|, |U_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})|, |S_i(P,\pi,\tau;\mathfrak{M})|)$, and the complete stage sequence $\mathcal{A}(P,\pi,\tau;\mathfrak{M}) = (\mathcal{A}_0,\ldots,\mathcal{A}_{m_{\mathrm{stage}}(P,\pi)})$; this record separates interface width, raw interface-state growth, reachability filtering, semantic quotienting, transition cost, reconstruction cost, verification cost, and output cost.

9.2.2 Comparison with Classical Baselines and Width Parameters

Every comparative claim requires a declared classical baseline. Let $N_{\mathrm{Classic}}(P, \tau; A_{\mathrm{Classic}})$ denote the complete elementary-operation count of a declared classical algorithm A_{Classic}, $T_{\mathrm{Classic}}(P, \tau; I_{\mathrm{Classic}})$ the measured wall-clock time of a declared classical implementation, and $T_{\mathrm{CPR}}(P, \pi, \tau; I_{\mathrm{CPR}}) > 0$ the measured wall-clock time of the declared CPR implementation. Assuming $N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) > 0$, the count-level gain and the measured wall-clock gain are

$$G_{\mathrm{count}}(P, \pi, \tau; \mathfrak{M}, A_{\mathrm{Classic}}) = \frac{N_{\mathrm{Classic}}(P, \tau; A_{\mathrm{Classic}})}{N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})},$$

$$G_{\mathrm{time}}(P, \pi, \tau; I_{\mathrm{CPR}}, I_{\mathrm{Classic}}) = \frac{T_{\mathrm{Classic}}(P, \tau; I_{\mathrm{Classic}})}{T_{\mathrm{CPR}}(P, \pi, \tau; I_{\mathrm{CPR}})};$$

the two gains remain distinct, $G_{\mathrm{count}} > 1 \not\Rightarrow G_{\mathrm{time}} > 1$. A valid comparison must fix the problem representation, the task, the required output, the correctness criterion, all preprocessing, the stopping condition, the classical algorithm, the CPR model, both implementations, the hardware and software environment, and the measurement protocol.

For graph-based or hypergraph-based benchmarks, the experimental record should include all available classical width parameters of the declared representation $G_P = \mathcal{G}(P)$, $\mathcal{W}(P, \pi, \tau; \mathfrak{M}, \mathcal{G}) = (\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}), \mathrm{tw}(G_P), \mathrm{pw}(G_P), \mathrm{bw}(G_P), \mathrm{cw}(G_P))$, not to infer a theorem from empirical correlation but to identify candidate upper bounds, candidate lower bounds, separation examples, ordering effects, and representation dependence – the experimental counterpart of the formal open problem in Section 9.1.5.

9.2.3 Boundary-Condition Audit

The CPR audit must distinguish finite instance records from uniform family-level certification. For one finite benchmark instance P, the instance-level audit record $\mathcal{A}_{\mathrm{BC}}^{\mathrm{inst}}(P) = (\sigma_1^{\mathrm{inst}}(P), \ldots, \sigma_4^{\mathrm{inst}}(P))$ has each component taking a value in $\{\mathrm{verified}, \mathrm{failed}, \mathrm{undetermined}\}$, recording respectively whether the required structural objects are explicitly available for the instance (BC1), whether the finite construction process and its measured operation count are completely recorded (BC2), whether soundness and completeness are verified for the declared finite task and benchmark (BC3), and whether the complete stage-indexed semantic-state and transition record is available (BC4). An instance-level record does not prove polynomial family behavior.

For a uniformly encoded family $\mathcal{F}$, the family-level certification record $\mathcal{A}_{\mathrm{BC}}^{\mathrm{fam}}(\mathcal{F}) = (\sigma_1^{\mathrm{fam}}(\mathcal{F}), \ldots, \sigma_4^{\mathrm{fam}}(\mathcal{F}))$ has each component taking a value in $\{\mathrm{proved}, \mathrm{disproved}, \mathrm{open}\}$; only the family-level record may support CPR-tractability certification. In particular, BC2 and BC4 are asymptotic uniform conditions and cannot be proved by one finite benchmark instance; BC4 controls the total number of stage-indexed semantic-state occurrences and the total number of reachable semantic transition edges, while the elementary operation cost of processing those states and edges is recorded separately in $N_{\mathrm{Transition}}$. If any family-level boundary condition remains open or is disproved, the family is not certified as CPR-tractable.

9.2.4 Benchmark Families and Ordering Ablation

A future benchmark corpus should contain favorable, neutral, and unfavorable instances, not only examples with visibly small candidate interfaces, so as to determine where CPR-WIDTH produces small active interfaces, where semantic quotienting is effective, where quotient construction dominates the runtime, where reconstruction or output dominates the runtime, and where CPR-WIDTH offers no advantage. Candidate benchmark groups include Boolean satisfiability instances with controlled clause-interaction structure; finite CSP instances with varying domain size, constraint arity, and primal-graph structure; graph-coloring instances with varying density and separator structure; arithmetic systems including full adders, ripple adders, and larger finite circuit blocks; tensor-structured systems with explicitly defined reconstruction components and active-index interfaces; routing and scheduling systems with bounded and unbounded interaction

boundaries; coding and decoding systems with controlled trellis structure; and negative-control instances designed to produce large active interfaces or expensive semantic signatures. For each benchmark family, the instance-generation procedure, input encoding, task, reconstruction model, admissibility rules, output requirements, baseline algorithms, and correctness criteria must be documented.

Because CPR-Width is ordering-dependent, experiments should compare multiple admissible orderings $\Pi_{\text{test}}(P, \tau; \mathfrak{M}) = \{\pi_1, \ldots, \pi_r\} \subseteq \Pi_{\text{adm}}(P, \tau; \mathfrak{M})$ while keeping every other model component fixed, recording for each π_j the tuple

$$(\widehat{\omega}_{\text{rec}}(P, \pi_j, \tau; \mathfrak{M}),\ \omega_{\text{rec}}(P, \pi_j, \tau; \mathfrak{M}),\ S_{\text{CPR}}(P, \pi_j, \tau; \mathfrak{M}),\ N_{\text{CPR}}(P, \pi_j, \tau; \mathfrak{M}),\ T_{\text{CPR}}(P, \pi_j, \tau; I_{\text{CPR}})).$$

Only the ordering may vary; the problem representation, task, reconstruction model, signature family, implementation, hardware, software environment, output requirement, and measurement protocol must remain fixed. A smaller width need not imply a smaller measured runtime: $\omega_{\text{rec}}(P, \pi_a, \tau; \mathfrak{M}) < \omega_{\text{rec}}(P, \pi_b, \tau; \mathfrak{M}) \not\Rightarrow T_{\text{CPR}}(P, \pi_a, \tau; I_{\text{CPR}}) < T_{\text{CPR}}(P, \pi_b, \tau; I_{\text{CPR}})$.

9.2.5 Evidence Classification and Future Research Sequence

CPR-WIDTH results should be classified by the specific evidence categories supporting them; the categories are not a single linear hierarchy, and a result may belong to more than one. The classification is: E0, Definition Evidence, a symbol, object, relation, operator, or structural quantity introduced formally; E1, Finite Verification Evidence, a claim checked exhaustively or directly on one explicitly stated finite example; E2, Structural Pattern Evidence, a recurring structural pattern identified across several models or examples; E3, Operation-Count Evidence, a finite or symbolic elementary-operation comparison relative to a declared baseline and accounting convention; E4, Asymptotic Theorem Evidence, a theorem proved uniformly for an encoded family under explicit assumptions and boundary conditions; and E5, Measured Runtime Evidence, a wall-clock claim supported by an implementation, hardware description, software environment, benchmark protocol, and reproducible measurements. The categories satisfy the non-implications E1 $\not\Rightarrow$ E4, E3 $\not\Rightarrow$ E5, E4 $\not\Rightarrow$ E5, and E5 $\not\Rightarrow$ E4: a finite verification does not establish an asymptotic theorem, an operation-count advantage does not establish measured acceleration, an asymptotic theorem does not establish a practical wall-clock advantage, and measured performance on a benchmark corpus does not establish a uniform asymptotic theorem. Every result should therefore state its complete evidence profile; a result may simultaneously possess E1, E3, and E5 support while having no E4 theorem.

On this basis, the realizations of Chapter 6 fall into three groups: mathematically fully verified (the full adder, by exhaustive finite verification, Chapter 8, and the ripple adder, by uniform proof for all n, Chapter 7); exemplarily verified (graph coloring on C_4, Section 6.3, Quantitative Verification for C_4: one concrete instance worked out in full, with no family-level certification); and theoretically formulated (Boolean satisfiability, constraint satisfaction, and tensor representations: structural candidate interfaces and reconstruction templates given, but no finite verification record or asymptotic theorem established for these classes in this manuscript).

A disciplined CPR-WIDTH research program may proceed through: complete exact mathematical characterization of CPR-Width for selected finite models; implementation of exact reachable-prefix, signature, interface, and semantic-quotient construction; recording BC1–BC4 evidence at the instance level; proving BC2 and BC4 at the uniform family level whenever an asymptotic tractability claim is intended; publishing complete component-wise operation-count records; comparing multiple admissible reconstruction orderings while keeping all other model and implementation conditions fixed; comparing CPR-WIDTH with declared classical baselines; extending finite analysis to uniformly encoded families; deriving formal relations among interface width, semantic-state growth, transition structure, and complete operation count; investigating exact and approximate relations to classical width parameters; determining controlled or constant CPR-Width bounds for additional problem-class subclasses; and conducting reproducible wall-clock experiments only after correctness, accounting rules, and output requirements have been fixed. This sequence prevents finite observations from being interpreted as asymptotic

theorems, and prevents structural state reduction from being interpreted as a complete runtime proof.

9.3 Final Conclusion

This chapter has combined the complexity-theoretic scope of CPR-WIDTH with its open mathematical, algorithmic, and experimental research program. The central structural-runtime relation remains $\boldsymbol{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \rightarrow \boldsymbol{S_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})} \rightarrow \boldsymbol{N_{\mathrm{CPR}}(P,\pi,\tau;\mathfrak{M})}$: the first arrow expresses the influence of the active reconstruction interface on the raw and semantic state spaces, the second the contribution of semantic-state processing to the complete CPR operation count, and neither arrow is a universal tractability theorem.

The principal unresolved questions concern efficient construction of favorable orderings; exact polynomial-time semantic encodings; efficient construction of minimum task-sufficient interfaces; determination of controlled or constant subclass-specific CPR-Width bounds; identification of nontrivial uniformly CPR-tractable families; comparison with classical width parameters; complete operation-count validation; and reproducible wall-clock benchmarking. The complete future tractability requirement is not merely $\mathrm{BC}_1 \wedge \mathrm{BC}_2 \wedge \mathrm{BC}_3 \wedge \mathrm{BC}_4$, but the full uniform quantified requirement stated at the start of this chapter; a polynomial-time conclusion is permitted only after one fixed decision task, one fixed uniform reconstruction model, one deterministic preprocessing algorithm, one deterministic ordering algorithm, one uniform correctness specification, and six fixed polynomial component bounds have been established for every encoded instance.

No NP-complete language is proved in this work to belong to $\mathbf{P}_{\mathrm{CPR}}$; accordingly, the contribution of CPR-WIDTH is not a proof of $\mathbf{P} = \mathbf{NP}$. Its contribution is a reconstruction-oriented mathematical language for separating active-interface width, reachable-state growth, semantic-state compression, transition cost, reconstruction cost, verification cost, output cost, operation-count gain, and measured runtime gain. Every conclusion must be limited to the specific evidence category or evidence profile supporting it: finite verification supports finite conclusions, structural patterns support structural conjectures and model comparisons, operation-count evidence supports count-level conclusions, asymptotic tractability requires a uniformly encoded family with all four boundary conditions and polynomial control of all six cost components, and measured acceleration requires a separate reproducible implementation and benchmark protocol.

Thus, CPR-WIDTH remains a formal framework for controlled partial reconstruction, semantic-state compression, and width-based runtime analysis. Its broader complexity-theoretic significance depends on the future identification of nontrivial uniformly encoded families satisfying the complete structural, semantic, algorithmic, and runtime requirements.

A Core Symbol Register

This appendix records the main symbols used throughout the manuscript, providing a compact reference for the CPR-WIDTH notation.

Table A.1: Core Symbol Register for CPR-WIDTH.

Symbol	Meaning
P	Finite computational problem or finite problem instance.
$V(P)$	Finite set of elementary components of P.
$v_1, \ldots, v_n$	Components of the finite problem representation.
π	Admissible reconstruction ordering.
$v_{\pi(i)}$	Component processed at reconstruction stage i.
V_i^{π}	Processed prefix after stage i.
$\overline{V}_i^{\pi}$	Unresolved suffix after stage i.
$\ell(P)$	Encoding length or input length of P.
B_i^{π}	Active reconstruction interface at stage i.
$B(P, \pi)$	Complete interface sequence induced by π.
$\omega_{\mathrm{rec}}(P, \pi)$	Controlled Partial Reconstruction Width induced by π.
$\omega_{\mathrm{rec}}(P)$	Minimum Controlled Partial Reconstruction Width over admissible orderings.
$\Pi_{\mathrm{adm}}(P)$	Set of admissible reconstruction orderings of P.
D_v	Finite local domain of component v.
$q(P)$	Maximum local domain size of P.
$\mathcal{U}_i(P, \pi)$	Raw interface-state space at stage i.
$\mathcal{U}_i^{\mathrm{reach}}(P, \pi)$	Reachable interface-state space at stage i.
$\mathcal{E}_i^{\tau}(u)$	Task-relative future extension set of state u.
$\equiv_{P,i}^{\tau}$	Future-extension equivalence relation at stage i.
$\mathcal{S}_i(P, \pi, \tau)$	Semantic quotient state space at stage i.
$S_{\mathrm{CPR}}(P, \pi, \tau)$	Maximum semantic-state count of the reconstruction process.
N_{CPR}	Total CPR operation count.
N_{Build}	Operations required to build the initial representation.
N_{Reduce}	Operations required for reduction and semantic quotienting.
$N_{\mathrm{Transition}}$	Operations required for stage-to-stage transitions.
$N_{\mathrm{Reconstruct}}$	Operations required to reconstruct admissible outputs.
N_{Verify}	Operations required to verify reconstructed outputs.
N_{Output}	Operations required to produce final outputs.
G_{count}	State-count or operation-count gain relative to a declared classical baseline.
G_{time}	Time gain relative to a declared classical runtime baseline.
$\mathcal{F}$	Uniformly encoded family of finite problem instances.
$L_{\mathcal{F}}$	Decision language associated with the family $\mathcal{F}$.
$\mathsf{P}_{\mathrm{CPR}}$	Class of languages certified by the CPR-WIDTH audit conditions.
P	Classical polynomial-time decision class.
NP	Classical nondeterministic polynomial-time decision class.

B Core Definition Register

This appendix summarizes the main definitions used in the manuscript.

Table B.1: Core definitions of CPR-WIDTH.

Definition / Object	Formula	Meaning
Finite component set	$V(P) = \{v_1, \ldots, v_n\}$	Finite set of elementary components of the problem instance P.
Admissible ordering	$\pi = (v_{\pi(1)}, \ldots, v_{\pi(n)})$	Order in which the finite components are processed.
Processed prefix	$V_i^\pi = \{v_{\pi(1)}, \ldots, v_{\pi(i)}\}$	Set of components processed after reconstruction stage i.
Unresolved suffix	$\overline{V}_i^\pi = V(P) \setminus V_i^\pi$	Set of components not yet processed after stage i.
Active reconstruction interface	$B_i^\pi \subseteq V_i^\pi$	Processed components whose information remains relevant for future reconstruction.
Interface sequence	$B(P,\pi) = (B_0^\pi, \ldots, B_n^\pi)$	Complete sequence of active interfaces induced by the ordering π.
Controlled Partial Reconstruction Width	$\omega_{\mathrm{rec}}(P,\pi) = \max\{0, \max_{0\le i\le n} \lvert B_i^\pi\rvert - 1\}$	Ordering-dependent normalized maximum active interface size.
Minimum CPR-Width	$\omega_{\mathrm{rec}}(P) = \min_{\pi\in\Pi_{\mathrm{adm}}(P)} \omega_{\mathrm{rec}}(P,\pi)$	Smallest CPR-Width attainable over all admissible reconstruction orderings.
Raw interface-state space	$\mathcal{U}_i(P,\pi) = \prod_{v\in B_i^\pi} D_v$	All possible assignments to the active interface at stage i.
Maximum local domain size	$q(P) = \max_{v\in V(P)} \lvert D_v\rvert$	Largest local domain size among the components of P.
Reachable interface states	$\mathcal{U}_i^{\mathrm{reach}}(P,\pi) \subseteq \mathcal{U}_i(P,\pi)$	Raw states that can occur under the declared reconstruction rules.
Future-extension equivalence	$u \equiv_{P,i}^{\tau} v \iff \mathcal{E}_i^{\tau}(u) = \mathcal{E}_i^{\tau}(v)$	Two reachable states are equivalent if they have the same task-relative admissible futures.
Semantic quotient	$\mathcal{S}_i(P,\pi,\tau) = \mathcal{U}_i^{\mathrm{reach}}(P,\pi)/\equiv_{P,i}^{\tau}$	State space obtained by merging reachable states with identical task-relevant futures.
Maximum semantic-state count	$S_{\mathrm{CPR}}(P,\pi,\tau) = \max_{0\le i\le n} \lvert\mathcal{S}_i(P,\pi,\tau)\rvert$	Maximum number of semantic quotient states over all reconstruction stages.
Binary semantic-state dimension	$D_{\mathrm{CPR}}(P,\pi) = \lceil \log_2 \max\{1, S_{\mathrm{CPR}}(P,\pi)\}\rceil$	Number of bits required to identify one semantic state.
Complete CPR operation count	$N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$	Complete operation-count model of the CPR process.

Continued on the next page

Definition / Object	**Formula**	**Meaning**
Post-transition cost	$R_{\text{post}} = N_{\text{Reconstruct}} + N_{\text{Verify}} + N_{\text{Output}}$	Combined reconstruction, verification, and output cost.
Count-level gain	$G_{\text{count}}(P,\pi) = \dfrac{N_{\text{Classic}}(P)}{N_{\text{CPR}}(P,\pi)}$	Operation-count comparison relative to a declared classical baseline.
Measured wall-clock gain	$G_{\text{time}}(P,\pi) = \dfrac{T_{\text{Classic}}(P)}{T_{\text{CPR}}(P,\pi)}$	Measured runtime comparison relative to a declared implementation baseline.
Certified CPR class	$L_{\mathcal{F}} \in \mathsf{P}_{\text{CPR}}$	The language is certified by a complete CPR audit model with polynomial operation count.
CPR tractability requirement	$\text{BC}_1 \wedge \text{BC}_2 \wedge \text{BC}_3 \wedge \text{BC}_4 \wedge N_{\text{CPR}}(P_x,\pi_x) \in \text{poly}(\ell(P_x))$	Complete condition required before a polynomial-time CPR conclusion is permitted.

C Proof Details

This appendix collects proof details for selected statements from the main text.

C.1 Interface Bound

Let $M(P,\pi) = \max_{0\le i\le n} |B_i^\pi|$. By definition, $\omega_{\mathrm{rec}}(P,\pi) = \max\{0, M(P,\pi)-1\}$. If $M(P,\pi) = 0$, all interfaces are empty, so $|B_i^\pi| = 0 \le \omega_{\mathrm{rec}}(P,\pi)+1$. If $M(P,\pi) \ge 1$, then $\omega_{\mathrm{rec}}(P,\pi) = M(P,\pi)-1$, so $\omega_{\mathrm{rec}}(P,\pi)+1 = M(P,\pi)$. Since every interface cardinality satisfies $|B_i^\pi| \le M(P,\pi)$, the interface bound follows.

C.2 Raw State Bound

By definition, $\mathcal{U}_i(P,\pi) = \prod_{v\in B_i^\pi} D_v$, so $|\mathcal{U}_i(P,\pi)| = \prod_{v\in B_i^\pi} |D_v|$. With $q(P) = \max_{v\in V(P)} |D_v|$, each factor satisfies $|D_v| \le q(P)$, hence $|\mathcal{U}_i(P,\pi)| \le q(P)^{|B_i^\pi|}$. Using the interface bound gives $|\mathcal{U}_i(P,\pi)| \le q(P)^{\omega_{\mathrm{rec}}(P,\pi)+1}$.

C.3 Reachable State Bound

The reachable interface state space is a subset of the raw interface state space, $\mathcal{U}_i^{\mathrm{reach}}(P,\pi) \subseteq \mathcal{U}_i(P,\pi)$, so $|\mathcal{U}_i^{\mathrm{reach}}(P,\pi)| \le |\mathcal{U}_i(P,\pi)|$. Using the raw state bound yields $|\mathcal{U}_i^{\mathrm{reach}}(P,\pi)| \le q(P)^{\omega_{\mathrm{rec}}(P,\pi)+1}$. Thus, reachable state growth is controlled by the same width bound as the raw interface state space.

C.4 Semantic-State Bound

The semantic quotient satisfies $|\mathcal{S}_i(P,\pi,\tau)| \le |\mathcal{U}_i^{\mathrm{reach}}(P,\pi)|$, and reachable states are a subset of raw states, $|\mathcal{U}_i^{\mathrm{reach}}(P,\pi)| \le |\mathcal{U}_i(P,\pi)|$; therefore $|\mathcal{S}_i(P,\pi,\tau)| \le |\mathcal{U}_i(P,\pi)| \le q(P)^{\omega_{\mathrm{rec}}(P,\pi)+1}$. Taking the maximum over all stages, $S_{\mathrm{CPR}}(P,\pi,\tau) = \max_{0\le i\le n} |\mathcal{S}_i(P,\pi,\tau)|$, hence $S_{\mathrm{CPR}}(P,\pi,\tau) \le q(P)^{\omega_{\mathrm{rec}}(P,\pi)+1}$.

C.5 Total Operation Bound

Assume that each operation component N_j is bounded by a polynomial $p_j(\ell(P))$. The total operation count is $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$, so $N_{\mathrm{CPR}} \le p_{\mathrm{Build}}(\ell(P)) + p_{\mathrm{Reduce}}(\ell(P)) + p_{\mathrm{Transition}}(\ell(P)) + p_{\mathrm{Reconstruct}}(\ell(P)) + p_{\mathrm{Verify}}(\ell(P)) + p_{\mathrm{Output}}(\ell(P))$. Since the sum of finitely many polynomials is again a polynomial, there exists a polynomial p_{CPR} such that $N_{\mathrm{CPR}} \le p_{\mathrm{CPR}}(\ell(P))$.

C.6 Quotient Preservation

Let u and v be reachable interface states at stage i with $u \equiv_{P,i}^{\tau} v$. By definition of future-extension equivalence, $\mathcal{E}_i^{\tau}(u) = \mathcal{E}_i^{\tau}(v)$, so every task-relevant admissible continuation represented by u is also represented by v, and conversely. Consequently, replacing reachable states by semantic equivalence classes does not remove any task-relevant future extension and does not introduce a new invalid future extension. Thus, quotienting by $\equiv_{P,i}^{\tau}$ preserves the declared task behavior.

D Audit Checklist

This appendix provides a compact checklist for applying the CPR-WIDTH audit model. The checklist records which declarations and boundary conditions must be present before a reconstruction process can be certified inside the CPR-WIDTH framework.

Table D.1: CPR-WIDTH audit checklist.

Audit item	**Required declaration or check**
Instance declaration	Declare the finite problem instance P, the component set $V(P)$, the local domains D_v, the encoding length $\ell(P)$, the declared task τ, and the intended output type.
Task declaration	Specify whether the task is decision, counting, optimization, enumeration, or another explicitly stated reconstruction objective.
Reconstruction ordering	Declare an admissible reconstruction ordering π. The ordering must be part of the audit because the width value depends on it.
Processed and unresolved parts	Declare the processed prefixes V_i^π and unresolved suffixes $\overline{V}_i^\pi$ for the reconstruction stages.
Active interfaces	Declare the active interfaces B_i^π and the full interface sequence $B(P,\pi)$.
Width declaration	Record the induced width $\omega_{\mathrm{rec}}(P,\pi) = \max\left\{0, \max_{0\le i\le n} \lvert B_i^\pi\rvert - 1\right\}$. If a bounded-width claim is made, the intended bound must be stated explicitly.
Raw state spaces	Declare the raw interface-state spaces $\mathcal{U}_i(P,\pi)$.
Reachable state spaces	Declare the reachable interface-state spaces $\mathcal{U}_i^{\mathrm{reach}}(P,\pi)$.
Future extensions	Declare the task-relative future extension sets $\mathcal{E}_i^\tau(u)$.
Semantic equivalence	Declare the equivalence relation $u \equiv_{P,i}^\tau v \iff \mathcal{E}_i^\tau(u) = \mathcal{E}_i^\tau(v)$.
Semantic quotient	Declare the semantic quotient spaces $\mathcal{S}_i(P,\pi,\tau)$.
Semantic-state count	Record $S_{\mathrm{CPR}}(P,\pi,\tau) = \max_{0\le i\le n} \lvert\mathcal{S}_i(P,\pi,\tau)\rvert$.
Operation-count components	Declare N_{Build}, N_{Reduce}, $N_{\mathrm{Transition}}$, $N_{\mathrm{Reconstruct}}$, N_{Verify}, and N_{Output}.
Complete operation count	Record the complete cost model $N_{\mathrm{CPR}} = N_{\mathrm{Build}} + N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} + N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}$.

Continued on the next page

Audit item	**Required declaration or check**
Polynomial bound check	For every component $j \in \{\text{Build}, \text{Reduce}, \text{Transition}, \text{Reconstruct}, \text{Verify}, \text{Output}\}$, there must exist a polynomial p_j such that $N_j(P, \pi, \tau) \leq p_j(\ell(P))$.
Soundness check	Verify that every reconstructed output is valid for the declared task: $y \in \text{Output}_{\text{CPR}}(P, \pi, \tau) \Longrightarrow \text{Valid}_\tau(P, y)$.
Completeness check	Verify that no required task-relevant solution is lost. For enumeration, one may require $\text{Sol}_\tau(P) \subseteq \text{Output}_{\text{CPR}}(P, \pi, \tau)$. For decision tasks, one may require $\text{Sol}_\tau(P) \neq \varnothing \Longrightarrow \text{CPRDecision}(P, \pi, \tau) = 1$.
Certification status	A positive answer to all required declarations and checks is needed for CPR-WIDTH certification.
Non-certification status	If any required item is unanswered, undetermined, or refuted, the audit does not certify the reconstruction process. This is an audit status, not a classical complexity conclusion.

The checklist is a control instrument. It does not prove tractability by itself. A polynomial-time conclusion requires all declared structural, semantic, correctness, and operation-count requirements to be satisfied for the relevant uniformly encoded family.

E One-Page Analysis Register

This appendix records the image pages used in the manuscript. The listed image pages are inserted through dedicated OPA insertion files and refer to existing image files in the Overleaf project. The mathematical notation displayed inside OPA images must be interpreted according to the formal definitions in the main text; if a visual label differs from a formal definition, the main-text definition takes priority.

Table E.1: One-Page Analysis Register for CPR-WIDTH.

Figure item	**Description**
OPA-SEM-01	Image file: `OPA-SEM-01.png`. Insertion file: `OPA-SEM-01.tex`. Location: Chapter 2. Purpose: Semantic-runtime architecture of CPR-WIDTH. The figure summarizes the transition from reconstruction width to semantic-state growth and the complete operation-count model.
OPA-CPR-01	Image file: `OPA-CPR-01.png`. Insertion file: `OPA-CPR-01.tex`. Location: Chapter 3. Purpose: Controlled partial reconstruction and semantic-state compression. The figure summarizes raw interface states, reachable states, semantic quotients, and the maximum semantic-state count.
OPA-RUN-01	Image file: `OPA-RUN-01.png`. Insertion file: `OPA-RUN-01.tex`. Location: Chapter 4. Purpose: Complete runtime framework of CPR-WIDTH. The figure summarizes the CPR pipeline, the classical baseline, the count-level gain, and the separation between count gain and measured runtime gain.
Cover image	Image file: `cover_image-4.png`. Insertion file: `cover_image-4.tex`. Location: Front cover. Purpose: Front cover image of CPR-WIDTH.
Required image files	The following image files must be present in the Overleaf project: `cover_image-4.png`, `OPA-SEM-01.png`, `OPA-CPR-01.png`, and `OPA-RUN-01.png`.
Required insertion files	The following insertion files must be present: `cover_image-4.tex`, `OPA-SEM-01.tex`, `OPA-CPR-01.tex`, and `OPA-RUN-01.tex`. All three OPA figures are activated in this version and inserted directly after their respective chapters.

The One-Page Analysis Register is a documentation appendix. It adds no independent mathematical claims beyond the main text.

Bibliography

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[2] Hans L. Bodlaender. "A Partial k-Arboretum of Graphs with Bounded Treewidth". In: *Theoretical Computer Science* 209.1–2 (1998), pp. 1–45.

[3] Stephen A. Cook. "The Complexity of Theorem-Proving Procedures". In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (1971), pp. 151–158.

[4] Marek Cygan et al. *Parameterized Algorithms*. Springer, 2015.

[5] Adnan Darwiche and Pierre Marquis. "A Knowledge Compilation Map". In: *Journal of Artificial Intelligence Research* 17 (2002), pp. 229–264.

[6] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[7] Reinhard Diestel. *Graph Theory*. 5th ed. Springer, 2017.

[8] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[9] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[10] David Money Harris and Sarah L. Harris. *Digital Design and Computer Architecture*. 2nd ed. Morgan Kaufmann, 2012.

[11] Richard M. Karp. "Reducibility Among Combinatorial Problems". In: *Complexity of Computer Computations* (1972), pp. 85–103.

[12] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2009), pp. 455–500.

[13] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[14] Neil Robertson and Paul D. Seymour. "Graph Minors. II. Algorithmic Aspects of Tree-Width". In: *Journal of Algorithms* 7.3 (1986), pp. 309–322.

[15] Francesca Rossi, Peter van Beek, and Toby Walsh, eds. *Handbook of Constraint Programming*. Elsevier, 2006.

[16] Michael Sipser. *Introduction to the Theory of Computation*. 3rd ed. Cengage Learning, 2012.