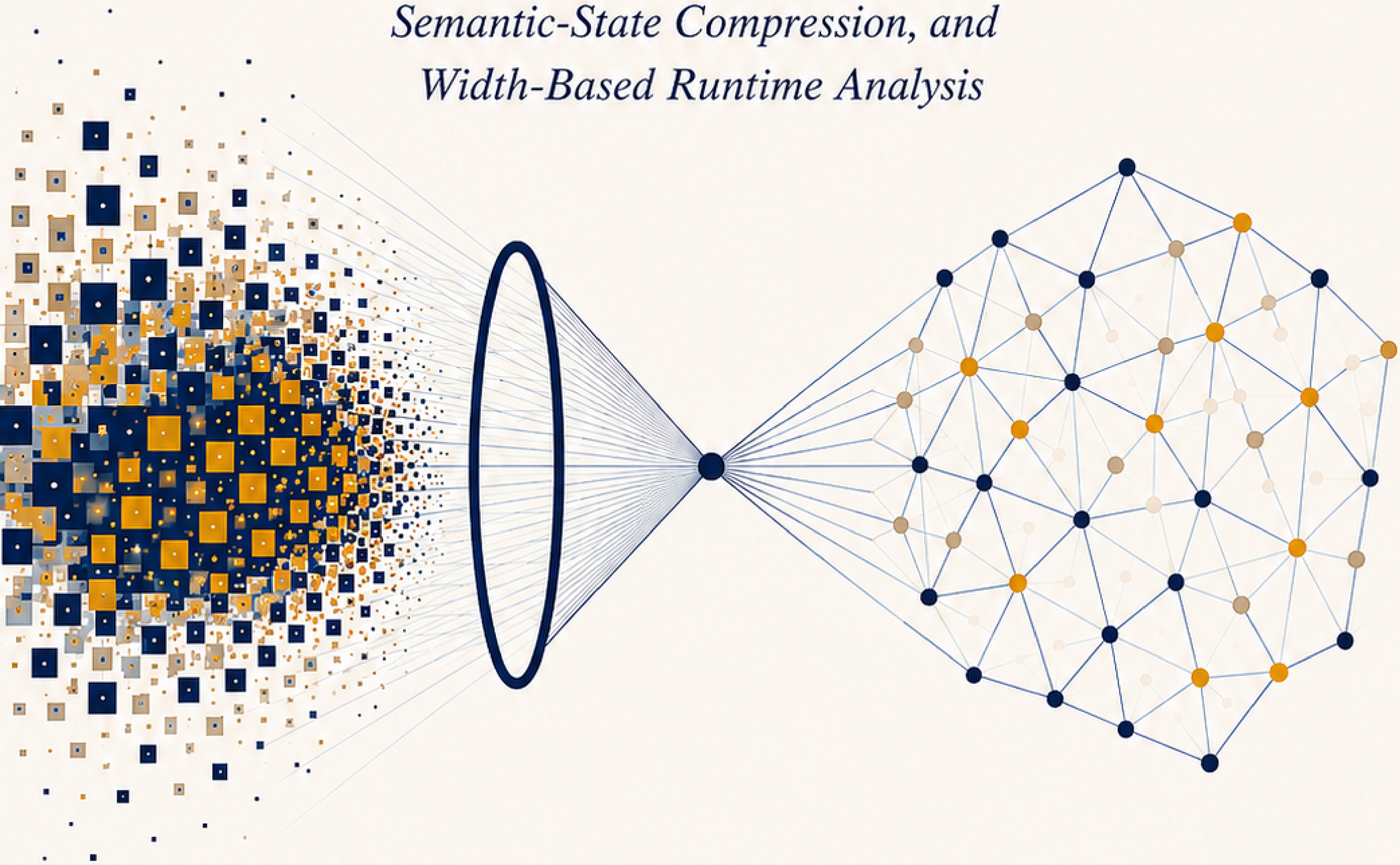
CPR-WIDTH

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-V1

Abstract

Controlled Partial Reconstruction Width (CPR-Width) is introduced as 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.

For a finite problem instance P and an admissible reconstruction ordering π, an active reconstruction interface records exactly those processed components whose information can still influence an admissible future continuation. CPR-Width is defined as the maximum cardinality of this interface during reconstruction and is minimized over all admissible orderings.

The framework distinguishes raw interface states, reachable states, and task-relative semantic states obtained through future-extension equivalence. This yields the structural chain

$$\omega_{\mathrm{rec}}(P,\pi) \longrightarrow S_{\mathrm{CPR}}(P,\pi) \longrightarrow N_{\mathrm{CPR}}(P,\pi), \tag{0.1}$$

which connects reconstruction width, semantic-state growth, and computational cost.

A complete runtime model separates construction, reduction, semantic transition, reconstruction, verification, and output costs. Four boundary conditions characterize explicit representation, polynomial constructibility, sound and complete reconstruction, and polynomial semantic-state control.

Representative realizations are presented for Boolean satisfiability, constraint satisfaction, graph coloring, tensor representations, and digital arithmetic. Exact finite verification is provided for the full adder and ripple-adder models.

The framework establishes a formal basis 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, and 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 problem is represented by the set of all admissible assignments, configurations, or certificates, and algorithmic complexity is analyzed with respect to the size of this global search space. From the viewpoint of reconstruction, however, this is often not the quantity that determines the essential computational difficulty. During a reconstruction process, previously processed information gradually loses its relevance: at every stage, some information may safely be discarded because it can no longer influence any admissible continuation, while other information must remain available until reconstruction is complete. This distinction between information that must remain active and information that may safely be discarded is fundamental.

The present manuscript studies the minimum number of processed components that must remain simultaneously active during a sound, complete, and task-relative reconstruction. Rather than measuring the size of the complete search space, the present theory measures this minimum active reconstruction interface, called the *Controlled Partial Reconstruction Width (CPR-WIDTH)*. Unlike classical graph-decomposition parameters such as treewidth and pathwidth, or the structural parameters of parameterized complexity, CPR-WIDTH is not defined on graphs or decomposition trees; it is a reconstruction-oriented notion whose value depends on the declared reconstruction model, the chosen ordering, the active interface, the declared task, and the reconstruction-safe future-signature family (related literature is discussed in Appendix E). Consequently, two procedures on the same structure may have different reconstruction widths whenever their active interfaces differ.

The objective of this manuscript is not to replace existing complexity theory, but to introduce a reconstruction-oriented structural parameter permitting a mathematical analysis of active information, semantic-state growth, and reconstruction complexity. The theory proceeds through three levels: the active interface, the reachable semantic-state space it generates, and the resulting computational effort. This yields the central structural chain

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

relating the maximal active interface, the corresponding semantic-state space, and the computational effort required by the declared reconstruction model; these quantities are developed progressively in Chapters 2–3.

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 goal is a rigorous theory of Controlled Partial Reconstruction Width and its structural, semantic, and computational consequences.

2 Controlled Partial Reconstruction Width

This chapter introduces the central structural parameter of the CPR-WIDTH framework.

The purpose is to determine 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:

$$\mathcal{R}_i^{\pi} \longrightarrow E_i^{\pi}(r) \longrightarrow \mathrm{Sig}_{\tau,i}^{\pi}(r) \longrightarrow B_i^{\pi,\tau} \longrightarrow \omega_{\mathrm{rec}}(P, \pi, \tau). \tag{2.1}$$

Thus, reachable prefix states and their task-relative 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

$$V(P) = \{v_1, \ldots, v_n\}. \tag{2.2}$$

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

$$\varnothing \neq D_v, \qquad |D_v| < \infty. \tag{2.3}$$

Let

$$q(P) = \max_{v \in V(P)} |D_v| \tag{2.4}$$

denote the maximum local domain size of P.

The complete assignment space is

$$\mathcal{X}(P) = \prod_{v \in V(P)} D_v. \tag{2.5}$$

Let

$$\mathrm{Sol}(P) \subseteq \mathcal{X}(P) \tag{2.6}$$

denote the set of complete feasible reconstructions.

A declared computational task is denoted by

$$\tau. \tag{2.7}$$

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

$$\mathfrak{M}(P,\tau) = \left(\mathrm{Adm}_{P,\tau}, \mathcal{R}, \Lambda, \delta, \mathrm{Sig}_{\tau}\right), \tag{2.8}$$

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;
- Sig_{τ} specifies a reconstruction-safe, task-relative future-signature family.

The components $\mathcal{R}$, Λ, δ, and Sig_{τ} denote parameterized families indexed jointly by the admissible ordering π and the reconstruction stage i — that is, $\mathcal{R} = (\mathcal{R}_i^{\pi}(P))_{\pi,i}$, $\Lambda = (\Lambda_i^{\pi})_{\pi,i}$, and $\delta = (\delta_i^{\pi})_{\pi,i}$ — rather than single fixed objects independent of the ordering.

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

Unless stated otherwise, every subsequent construction is understood relative to these fixed declarations.

2.2 Admissible Reconstruction Orderings

A reconstruction ordering is a permutation

$$\pi = \Big(v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}\Big). \tag{2.9}$$

Definition 2.2 (Admissible Reconstruction Ordering)**.** An ordering π is admissible for the model $\mathfrak{M}(P,\tau)$ if

$$\mathrm{Adm}_{P,\tau}(\pi) = 1. \tag{2.10}$$

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

$$\Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M}) = \left\{\pi \in \mathrm{Sym}(V(P)) : \mathrm{Adm}_{P,\tau}(\pi) = 1\right\}. \tag{2.11}$$

Assume throughout that

$$\Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M}) \neq \varnothing. \tag{2.12}$$

When both the task τ and reconstruction model $\mathfrak{M}$ are fixed throughout a section, the shorter notation

$$\Pi_{\mathrm{adm}}(P) \tag{2.13}$$

may be used as a local abbreviation.

2.3 Processed Prefix and Unresolved Suffix

Let

$$\pi \in \Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M}) \tag{2.14}$$

be fixed.

After reconstruction stage i, the processed prefix is

$$V_i^\pi = \{v_{\pi(1)}, \ldots, v_{\pi(i)}\}, \qquad 0 \leq i \leq n. \tag{2.15}$$

The unresolved suffix is

$$\overline{V_i^\pi} = V(P) \setminus V_i^\pi. \tag{2.16}$$

The boundary cases are

$$V_0^\pi = \varnothing, \qquad V_n^\pi = V(P), \tag{2.17}$$

and

$$\overline{V_0^\pi} = V(P), \qquad \overline{V_n^\pi} = \varnothing. \tag{2.18}$$

The complete prefix-assignment space at stage i is

$$\mathcal{A}_i^\pi(P) = \prod_{v \in V_i^\pi} D_v. \tag{2.19}$$

The empty Cartesian product is interpreted as the singleton containing the empty assignment:

$$\mathcal{A}_0^\pi(P) = \{\varnothing\}. \tag{2.20}$$

The complete suffix-assignment space at stage i is

$$\mathcal{C}_i^\pi(P) = \prod_{v \in \overline{V_i^\pi}} D_v. \tag{2.21}$$

Accordingly,

$$\mathcal{C}_n^\pi(P) = \{\varnothing\}. \tag{2.22}$$

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.

Reachability is determined exclusively by the declared transition system and not by global feasibility.

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

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

Since $V_i^\pi \cap \overline{V_i^\pi} = \varnothing$, the union $r \cup s$ is a well-defined complete assignment.

At a fixed reconstruction stage i, all sets $E_i^\pi(r)$ are subsets of the same continuation universe $\mathcal{C}_i^\pi(P)$.

The continuation set itself is independent of the computational task. Task dependence enters only through the evaluation map.

The declared task τ determines a stage-relative evaluation map

$$\Theta_{\tau,i}^\pi : \mathcal{R}_i^\pi(P) \times \mathcal{P}\Big(\mathcal{C}_i^\pi(P)\Big) \longrightarrow Y_{\tau,i}.$$

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 a counting task, one may use $\mathrm{Ans}_{\mathrm{count},i}^\pi(r) = |E_i^\pi(r)|$.

For an enumeration task, one may use $\mathrm{Ans}_{\mathrm{enum},i}^\pi(r) = \{r \cup s : s \in E_i^\pi(r)\}$.

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

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

For every admissible next value $a \in \Lambda_i^\pi(r)$, the transition map is

$$\delta_i^\pi(r, a) = r \cup \{w_i^\pi \mapsto a\}. \tag{2.23}$$

Thus,

$$\delta_i^\pi(r, a) \in \mathcal{R}_{i+1}^\pi(P). \tag{2.24}$$

2.7 Reconstruction-Safe Future Signatures

A task value alone may be too coarse for a stage-wise reconstruction process.

For example, 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.

For every admissible ordering and reconstruction stage, let $\Sigma_{\tau,i}^\pi$ denote the abstract signature space associated with the declared reconstruction model.

Definition 2.4 (Reconstruction-Safe Future-Signature Family)**.** A family of maps

$$\mathrm{Sig}_{\tau,i}^\pi : \mathcal{R}_i^\pi(P) \longrightarrow \Sigma_{\tau,i}^\pi, \qquad 0 \le i \le n, \tag{2.25}$$

is called reconstruction-safe if the following conditions hold.

S1 — Task preservation.
For all reachable states $r, r' \in \mathcal{R}_i^\pi(P)$,

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r') \tag{2.26}$$

implies

$$\mathrm{Ans}_{\tau,i}^\pi(r) = \mathrm{Ans}_{\tau,i}^\pi(r'). \tag{2.27}$$

S2 — Preservation of admissible next values.

For every $i < n$,

$$\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r') \tag{2.28}$$

implies

$$\Lambda^{\pi}_{i}(r) = \Lambda^{\pi}_{i}(r'). \tag{2.29}$$

S3 — Successor congruence.

If

$$\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r') \tag{2.30}$$

and

$$a \in \Lambda^{\pi}_{i}(r) = \Lambda^{\pi}_{i}(r'), \tag{2.31}$$

then

$$\mathrm{Sig}^{\pi}_{\tau,i+1}\Big(\delta^{\pi}_{i}(r,a)\Big) = \mathrm{Sig}^{\pi}_{\tau,i+1}\Big(\delta^{\pi}_{i}(r',a)\Big). \tag{2.32}$$

The reconstruction-safe signature therefore defines a task-relative transition congruence rather than merely an equality relation over feasible continuation sets.

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

$$\mathrm{Sig}^{\pi}_{\tau,i}(r) = r \tag{2.33}$$

is always reconstruction-safe.

Therefore, 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

$$B \subseteq V_i^\pi \tag{2.34}$$

is called a task-sufficient reconstruction interface if

$$\forall r, r' \in \mathcal{R}_i^\pi(P): \qquad r|_B = r'|_B \Longrightarrow \mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r'). \tag{2.35}$$

This condition guarantees that the complete reconstruction-safe future signature is determined by the values retained on B.

Therefore, 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

$$\mathfrak{B}_i^{\pi,\tau}(P) = \left\{ B \subseteq V_i^\pi : \begin{array}{l} \forall r, r' \in \mathcal{R}_i^\pi(P), \\ r|_B = r'|_B \Rightarrow \mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r') \end{array} \right\}. \tag{2.36}$$

Lemma 2.1 (Existence of a Task-Sufficient Interface)**.** *For every reconstruction stage i,*

$$\mathfrak{B}_i^{\pi,\tau}(P) \neq \varnothing. \tag{2.37}$$

Proof. The complete processed prefix V_i^π is task-sufficient.

Indeed, if

$$r|_{V_i^\pi} = r'|_{V_i^\pi}, \tag{2.38}$$

then

$$r = r'. \tag{2.39}$$

Consequently,

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r'). \tag{2.40}$$

Therefore,

$$V_i^\pi \in \mathfrak{B}_i^{\pi,\tau}(P). \tag{2.41}$$

□

2.9 Interface Factorization

For

$$B \subseteq V_i^\pi, \tag{2.42}$$

define the coordinate projection

$$\rho^{\pi}_{i,B} : \mathcal{R}^{\pi}_{i}(P) \longrightarrow \prod_{v\in B} D_v \tag{2.43}$$

by

$$\rho^{\pi}_{i,B}(r) = r|_B. \tag{2.44}$$

Proposition 2.1 (Interface Factorization)**.** *A set $B \subseteq V^{\pi}_{i}$ is task-sufficient if and only if there exists a unique map on the projected reachable interface-state space*

$$\overline{\mathrm{Sig}}^{\pi}_{\tau,i,B} : \rho^{\pi}_{i,B}\Big(\mathcal{R}^{\pi}_{i}(P)\Big) \longrightarrow \Sigma^{\pi}_{\tau,i} \tag{2.45}$$

such that

$$\mathrm{Sig}^{\pi}_{\tau,i} = \overline{\mathrm{Sig}}^{\pi}_{\tau,i,B} \circ \rho^{\pi}_{i,B}. \tag{2.46}$$

Proof. Assume first that B is task-sufficient.
For

$$z \in \rho^{\pi}_{i,B}\Big(\mathcal{R}^{\pi}_{i}(P)\Big), \tag{2.47}$$

choose a reachable state r satisfying

$$\rho^{\pi}_{i,B}(r) = z \tag{2.48}$$

and define

$$\overline{\mathrm{Sig}}^{\pi}_{\tau,i,B}(z) = \mathrm{Sig}^{\pi}_{\tau,i}(r). \tag{2.49}$$

If r' is another reachable state satisfying

$$\rho^{\pi}_{i,B}(r') = z, \tag{2.50}$$

then

$$r|_B = r'|_B. \tag{2.51}$$

Task sufficiency gives

$$\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r'). \tag{2.52}$$

Hence, the map is well defined: the definition is independent of the chosen representative.

Conversely, assume that the factorization exists.

If

$$r|_B = r'|_B, \tag{2.53}$$

then

$$\rho^{\pi}_{i,B}(r) = \rho^{\pi}_{i,B}(r'). \tag{2.54}$$

Therefore,

$$\begin{aligned}
\mathrm{Sig}^{\pi}_{\tau,i}(r) &= \overline{\mathrm{Sig}}^{\pi}_{\tau,i,B}\Big(\rho^{\pi}_{i,B}(r)\Big) && (2.55)\\
&= \overline{\mathrm{Sig}}^{\pi}_{\tau,i,B}\Big(\rho^{\pi}_{i,B}(r')\Big) && (2.56)\\
&= \mathrm{Sig}^{\pi}_{\tau,i}(r'). && (2.57)
\end{aligned}$$

Thus, B is task-sufficient.

Uniqueness follows because every element of the domain is the projection of at least one reachable prefix state.

□

Corollary 2.1 (Well-Defined Admissible Next-Value Set on the Interface)**.** *Let $B = B_i^{\pi,\tau}$ be a task-sufficient active interface at a nonterminal stage $i < n$. For $u \in \rho^{\pi}_{i,B}\Big(\mathcal{R}^{\pi}_i(P)\Big)$, define*

$$\widehat{\Lambda}_i^{\pi,\tau}(u) = \Lambda_i^{\pi}(r). \tag{2.58}$$

for any $r \in \mathcal{R}^{\pi}_i(P)$ with $r|_B = u$. This is independent of the choice of r.

Proof. Let $r, r' \in \mathcal{R}^{\pi}_i(P)$ satisfy $r|_B = r'|_B = u$. By the Interface Factorization Proposition, $\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r')$. By S2 (Preservation of Admissible Next Values), $\Lambda_i^{\pi}(r) = \Lambda_i^{\pi}(r')$.

□

The set $\widehat{\Lambda}_i^{\pi,\tau}(u)$ is therefore a well-defined function of the interface state u alone, independently of which reachable prefix state projects onto it.

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

$$b_i^{\pi,\tau}(P) = \min\left\{|B| : B \in \mathfrak{B}_i^{\pi,\tau}(P)\right\}. \tag{2.59}$$

The minimum exists because

$$\mathfrak{B}_i^{\pi,\tau}(P) \neq \varnothing \tag{2.60}$$

and V_i^π is finite.

One may select any cardinality-minimal task-sufficient set as the active reconstruction interface:

$$B_i^{\pi,\tau} \in \operatorname*{arg\,min}_{B \in \mathfrak{B}_i^{\pi,\tau}(P)} |B|. \tag{2.61}$$

The minimizing set need not be unique, but every minimizing set has the same cardinality:

$$|B_i^{\pi,\tau}| = b_i^{\pi,\tau}(P). \tag{2.62}$$

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.

Whenever multiple cardinality-minimal interfaces exist, $B_i^{\pi,\tau}$ denotes one selected representative from the corresponding $\arg\min$ family.

The complete selected interface sequence is

$$\mathcal{B}(P, \pi, \tau) = \left(B_0^{\pi,\tau}, B_1^{\pi,\tau}, \dots, B_n^{\pi,\tau}\right). \tag{2.63}$$

Since

$$\mathcal{R}_0^\pi(P) = \{\varnothing\}, \tag{2.64}$$

one has

$$b_0^{\pi,\tau}(P) = 0 \tag{2.65}$$

and the initial interface may be selected as

$$B_0^{\pi,\tau} = \varnothing. \tag{2.66}$$

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 the 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

$$r, r' \in \mathcal{R}_i^\pi(P) \tag{2.67}$$

such that

$$r|_{V_i^\pi \setminus \{v\}} = r'|_{V_i^\pi \setminus \{v\}}, \tag{2.68}$$

but

$$\mathrm{Sig}_{\tau,i}^\pi(r) \neq \mathrm{Sig}_{\tau,i}^\pi(r'). \tag{2.69}$$

Then every task-sufficient interface contains v.

Proof. Let B be task-sufficient and suppose that

$$v \notin B. \tag{2.70}$$

Then

$$B \subseteq V_i^\pi \setminus \{v\}. \tag{2.71}$$

Consequently,

$$r|_B = r'|_B. \tag{2.72}$$

Task sufficiency would imply

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r'), \tag{2.73}$$

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

$$\mathcal{R}_i^\pi(P) = \{00, 11\}, \tag{2.74}$$

with

$$\mathrm{Sig}^{\pi}_{\tau,i}(00) \neq \mathrm{Sig}^{\pi}_{\tau,i}(11). \tag{2.75}$$

There are no two reachable states that differ only in a, and there are no two reachable states 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

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = \max_{0 \leq i \leq n} b_i^{\pi,\tau}(P). \tag{2.76}$$

The value

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \tag{2.77}$$

is the actual minimum number of components that must remain simultaneously active at the widest reconstruction stage.

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

$$\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = \max\left\{0,\ \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) - 1\right\}. \tag{2.78}$$

The subtraction by one is an optional normalization convention chosen to facilitate comparison with classical width parameters, including treewidth and pathwidth. The primary CPR invariant is the unnormalized quantity $\widehat{\omega}_{\mathrm{rec}}$.

It is not forced by the reconstruction model.

Accordingly,

$$\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = 0 \tag{2.79}$$

may still mean that one component must remain active.

The unnormalized quantity $\widehat{\omega}_{\mathrm{rec}}$ is the primary structural invariant, whereas ω_{rec} is a derived normalized convention.

When τ and $\mathfrak{M}$ are fixed, the shorter forms

$$\widehat{\omega}_{\text{rec}}(P, \pi) \tag{2.80}$$

and

$$\omega_{\text{rec}}(P, \pi) \tag{2.81}$$

may be used.

2.13 Interface Bound

Lemma 2.3 (Interface Bound)**.** *For every stage i,*

$$b_i^{\pi,\tau}(P) \leq \widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}). \tag{2.82}$$

Moreover,

$$\widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) \leq \omega_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) + 1, \tag{2.83}$$

with equality whenever $\widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) \geq 1$; the inequality is strict only at the boundary case $\widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) = 0$.
Consequently, for every selected minimum active interface,

$$|B_i^{\pi,\tau}| \leq \omega_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) + 1. \tag{2.84}$$

Proof. The first inequality follows immediately from

$$\widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) = \max_{0 \leq j \leq n} b_j^{\pi,\tau}(P). \tag{2.85}$$

Let

$$M = \widehat{\omega}_{\text{rec}}(P, \pi, \tau; \mathfrak{M}). \tag{2.86}$$

If $M = 0$, then by definition

$$\omega_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) = 0. \tag{2.87}$$

Hence,

$$M = 0 \leq \omega_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) + 1. \tag{2.88}$$

If $M \geq 1$, then

$$\omega_{\text{rec}}(P, \pi, \tau; \mathfrak{M}) = M - 1, \tag{2.89}$$

and therefore

$$M = \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1. \tag{2.90}$$

Finally,

$$|B_i^{\pi,\tau}| = b_i^{\pi,\tau}(P), \tag{2.91}$$

because $B_i^{\pi,\tau}$ is cardinality-minimal.

□

2.14 Minimum CPR-Width over Admissible Orderings

Because

$$\Pi_{\mathrm{adm}}(P, \tau; \mathfrak{M}) \neq \varnothing, \tag{2.92}$$

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

$$\omega_{\mathrm{rec}}(P, \tau; \mathfrak{M}) = \min_{\pi \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})} \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}). \tag{2.93}$$

The corresponding unnormalized minimum is

$$\widehat{\omega}_{\mathrm{rec}}(P, \tau; \mathfrak{M}) = \min_{\pi \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M})} \widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}). \tag{2.94}$$

When the task and reconstruction model are fixed, these quantities may be written as

$$\omega_{\mathrm{rec}}(P) \tag{2.95}$$

and

$$\widehat{\omega}_{\mathrm{rec}}(P). \tag{2.96}$$

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*

$$\omega_{\text{rec}}(P, \pi_1, \tau; \mathfrak{M}) \neq \omega_{\text{rec}}(P, \pi_2, \tau; \mathfrak{M}). \tag{2.97}$$

Proof. Let

$$V(P) = \{a, b, c\} \tag{2.98}$$

with binary domains

$$D_a = D_b = D_c = \{0, 1\}. \tag{2.99}$$

Impose the constraints

$$a = c \tag{2.100}$$

and

$$b = c. \tag{2.101}$$

Thus,

$$\mathrm{Sol}(P) = \{000, 111\}. \tag{2.102}$$

Let the declared task be decision.

A partial assignment is reachable if every constraint whose complete scope has already been processed is satisfied.

Reachability in this model therefore refers to transition admissibility and not to existence of a complete feasible extension.

Constraints involving unprocessed components are not evaluated at this stage; therefore a reachable prefix may later become infeasible after additional components are reconstructed.

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

$$\pi_1 = (a, b, c). \tag{2.103}$$

After stage 2, neither equality constraint has been completely processed, because both contain c.

Therefore,

$$\mathcal{R}_2^{\pi_1}(P) = \{00, 01, 10, 11\}. \tag{2.104}$$

The admissible next-value sets for c are

$$\Lambda_2^{\pi_1}(00) = \{0\}, \tag{2.105}$$

$$\Lambda_2^{\pi_1}(11) = \{1\}, \tag{2.106}$$

and

$$\Lambda_2^{\pi_1}(01) = \Lambda_2^{\pi_1}(10) = \varnothing. \tag{2.107}$$

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,

$$b_2^{\pi_1,\mathrm{dec}}(P) = 2. \tag{2.108}$$

At all other stages, an interface of size at most one is sufficient. Therefore,

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) = 2 \tag{2.109}$$

and

$$\omega_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) = 1. \tag{2.110}$$

Now consider the ordering

$$\pi_2 = (a, c, b). \tag{2.111}$$

After stage 2, the constraint $a = c$ has already been processed. The inconsistent prefixes are therefore removed, and

$$\mathcal{R}_2^{\pi_2}(P) = \{00, 11\}. \tag{2.112}$$

The admissible next-value sets for b are

$$\Lambda_2^{\pi_2}(00) = \{0\} \tag{2.113}$$

and

$$\Lambda_2^{\pi_2}(11) = \{1\}. \tag{2.114}$$

Either component a or component c distinguishes these two reachable states. Thus,

$$b_2^{\pi_2,\mathrm{dec}}(P) = 1. \tag{2.115}$$

At every stage, an interface of size at most one is task-sufficient. Hence,

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M}) = 1 \tag{2.116}$$

and

$$\omega_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M}) = 0. \tag{2.117}$$

Therefore,

$$\omega_{\mathrm{rec}}(P, \pi_1, \mathrm{dec}; \mathfrak{M}) \neq \omega_{\mathrm{rec}}(P, \pi_2, \mathrm{dec}; \mathfrak{M}). \tag{2.118}$$

□

2.16 Structural Interpretation

Controlled Partial Reconstruction Width does not measure the total size of the complete candidate space.

The unnormalized quantity

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \tag{2.119}$$

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

$$\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \tag{2.120}$$

is obtained by the adopted minus-one convention.

A task-sufficient interface satisfies

$$r|_B = r'|_B \Longrightarrow \mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r'). \tag{2.121}$$

Equivalently, the future-signature map factors through the interface projection:

$$\mathrm{Sig}^{\pi}_{\tau,i} = \overline{\mathrm{Sig}}^{\pi}_{\tau,i,B} \circ \rho^{\pi}_{i,B}. \tag{2.122}$$

This factorization is the formal statement that all jointly forgotten components are semantically and transitionally unnecessary for the declared reconstruction task.

CPR-Width is therefore dependent on

- the component representation;
- the component domains;
- the reconstruction 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,

$$B_i^{\pi,\mathrm{dec}} \neq B_i^{\pi,\mathrm{count}} \tag{2.123}$$

and

$$\omega_{\mathrm{rec}}(P, \pi, \mathrm{dec}; \mathfrak{M}) \neq \omega_{\mathrm{rec}}(P, \pi, \mathrm{count}; \mathfrak{M}) \tag{2.124}$$

may occur.

CPR-Width counts active components. It does not directly measure their bit-level information content.

Consequently, CPR-Width is invariant under any representation-preserving renaming of component identifiers.

Local domain sizes, projected interface-state spaces, reachable interface states, and semantic quotient states are developed in the following chapter.

Chapter Summary

The raw prefix-state space $\mathcal{A}_i^{\pi}(P)$ is first restricted to the reachable prefix-state space $\mathcal{R}_i^{\pi}(P)$ by the declared admissibility and transition rules. The reconstruction-safe future-signature map $\mathrm{Sig}^{\pi}_{\tau,i}$ then identifies reachable states that are indistinguishable for the declared task and admissible transitions under conditions S1–S3. This yields a task-sufficient active interface $B_i^{\pi;\tau}$ of minimum cardinality.

The unnormalized width $\widehat{\omega}_{\mathrm{rec}}(P, \pi)$ records the largest minimum interface cardinality encountered during reconstruction, whereas $\omega_{\mathrm{rec}}(P, \pi)$ denotes its adopted normalized form. The semantic-state count $S_{\mathrm{CPR}}(P, \pi; \tau)$, formally introduced in Chapter 3, records the largest number of task-relevant semantic classes encountered along the same reconstruction process.

Neither quantity requires literal equality of continuation sets. Instead, the declared task value, the admissible next-value sets, and the successor-signature structure must be preserved. The following chapter extends this structural analysis to the complete operation-count quantity $N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})$.

3 Semantic State Compression

This chapter develops the semantic-state compression layer of the CPR-WIDTH framework. Building on the reconstruction-safe future-signature family and the minimum active interfaces introduced in Chapter 2, the chapter defines reachable interface states, task-relative semantic equivalence, semantic quotient spaces, quotient transitions, and the resulting semantic-state bounds. The construction proceeds through the chain

$$\mathcal{R}_i^{\pi}(P) \longrightarrow \mathcal{U}_i^{\mathrm{reach}}(P,\pi) \longrightarrow \mathcal{S}_i^{\tau}(P,\pi) \longrightarrow S_{\mathrm{CPR}}(P,\pi;\tau). \tag{3.1}$$

Thus, semantic compression is derived from the reconstruction-safe structure of Chapter 2 and does not introduce an independent or weaker future-signature system.

3.1 Interface States and Induced Future Signatures

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

$$\pi \in \Pi_{\mathrm{adm}}(P,\tau;\mathfrak{M}) \tag{3.2}$$

be an admissible reconstruction ordering. For every reconstruction stage i, let

$$B_i^{\pi,\tau} \subseteq V_i^{\pi} \tag{3.3}$$

be one selected cardinality-minimal task-sufficient interface as defined in Chapter 2. The raw interface-state space is

$$\mathcal{U}_i(P,\pi,\tau;\mathfrak{M}) = \prod_{v\in B_i^{\pi,\tau}} D_v. \tag{3.4}$$

When the task τ and reconstruction model $\mathfrak{M}$ are fixed, the shorter notation

$$\mathcal{U}_i(P,\pi) \tag{3.5}$$

may be used. The coordinate projection from reachable prefix states to interface states is

$$\rho_i^{\pi,\tau} : \mathcal{R}_i^{\pi}(P) \longrightarrow \mathcal{U}_i(P,\pi,\tau;\mathfrak{M}), \qquad \rho_i^{\pi,\tau}(r) = r|_{B_i^{\pi,\tau}}. \tag{3.6}$$

Definition 3.1 (Reachable Interface-State Space)**.** The reachable interface-state space at reconstruction stage i is

$$\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) = \rho_i^{\pi,\tau}\Big(\mathcal{R}_i^{\pi}(P)\Big). \tag{3.7}$$

When τ and $\mathfrak{M}$ are fixed, this space may be written as

$$\mathcal{U}_i^{\mathrm{reach}}(P,\pi). \tag{3.8}$$

Thus,

$$\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \subseteq \mathcal{U}_i(P,\pi,\tau;\mathfrak{M}). \tag{3.9}$$

By the Interface Factorization Proposition of Chapter 2, the reconstruction-safe future-signature map

$$\mathrm{Sig}^{\pi}_{\tau,i} : \mathcal{R}^{\pi}_{i}(P) \longrightarrow \Sigma^{\pi}_{\tau,i} \tag{3.10}$$

factors through the projection onto every task-sufficient interface. In particular, for the selected minimum interface $B^{\pi,\tau}_{i}$, there exists a unique map

$$\overline{\mathrm{Sig}}^{\pi}_{\tau,i} : \mathcal{U}^{\mathrm{reach}}_{i}(P, \pi, \tau; \mathfrak{M}) \longrightarrow \Sigma^{\pi}_{\tau,i} \tag{3.11}$$

such that

$$\mathrm{Sig}^{\pi}_{\tau,i} = \overline{\mathrm{Sig}}^{\pi}_{\tau,i} \circ \rho^{\pi,\tau}_{i}. \tag{3.12}$$

Definition 3.2 (Induced Future Signature)**.** For every reachable interface state

$$u \in \mathcal{U}^{\mathrm{reach}}_{i}(P, \pi, \tau; \mathfrak{M}), \tag{3.13}$$

the induced future signature is defined by

$$\widehat{\sigma}^{\pi,\tau}_{P,i}(u) = \overline{\mathrm{Sig}}^{\pi}_{\tau,i}(u). \tag{3.14}$$

Thus,

$$\widehat{\sigma}^{\pi,\tau}_{P,i} : \mathcal{U}^{\mathrm{reach}}_{i}(P, \pi, \tau; \mathfrak{M}) \longrightarrow \Sigma^{\pi}_{\tau,i}. \tag{3.15}$$

Equivalently, if

$$u = r|_{B^{\pi,\tau}_{i}} \tag{3.16}$$

for some reachable prefix state $r \in \mathcal{R}^{\pi}_{i}(P)$, then

$$\widehat{\sigma}^{\pi,\tau}_{P,i}(u) = \mathrm{Sig}^{\pi}_{\tau,i}(r). \tag{3.17}$$

This definition is independent of the selected reachable representative r. Indeed, if

$$r|_{B^{\pi,\tau}_{i}} = r'|_{B^{\pi,\tau}_{i}}, \tag{3.18}$$

then task sufficiency implies

$$\mathrm{Sig}^{\pi}_{\tau,i}(r) = \mathrm{Sig}^{\pi}_{\tau,i}(r'). \tag{3.19}$$

The induced signature therefore preserves all three reconstruction-safe conditions established in Chapter 2:

- preservation of the declared task-relative future value;
- preservation of admissible next reconstruction values;
- successor congruence under admissible transitions.

No new or weaker signature system is introduced in this chapter.

3.2 Task-Relative Signature Equivalence

Throughout this section, P, π, τ, $\mathfrak{M}$, and the reconstruction stage i are fixed.

Definition 3.3 (Task-Relative Signature Equivalence)**.** For reachable interface states

$$u, v \in \mathcal{U}^{\mathrm{reach}}_{i}(P, \pi, \tau; \mathfrak{M}), \tag{3.20}$$

define

$$u \equiv^{\pi,\tau}_{P,i} v \quad \Longleftrightarrow \quad \widehat{\sigma}^{\pi,\tau}_{P,i}(u) = \widehat{\sigma}^{\pi,\tau}_{P,i}(v). \tag{3.21}$$

When the ordering, task, and reconstruction model are fixed throughout the discussion, the shorter notation

$$u \equiv_{P,i} v \tag{3.22}$$

may be used.

The relation compares reconstruction-safe task-relative signatures. It does not require literal equality of complete feasible continuation sets.

Proposition 3.1 (Signature Equivalence Is an Equivalence Relation)**.** *For every fixed problem instance P, admissible reconstruction ordering π, stage i, declared task τ, and reconstruction model $\mathfrak{M}$, the relation*

$$\equiv_{P,i}^{\pi,\tau} \tag{3.23}$$

is an equivalence relation on

$$\mathcal{U}_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M}). \tag{3.24}$$

Proof. For every reachable interface state u,

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(u), \tag{3.25}$$

so the relation is reflexive. If

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v), \tag{3.26}$$

then

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(v) = \widehat{\sigma}_{P,i}^{\pi,\tau}(u), \tag{3.27}$$

so the relation is symmetric. Finally, if

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v) \tag{3.28}$$

and

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(v) = \widehat{\sigma}_{P,i}^{\pi,\tau}(w), \tag{3.29}$$

then

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(w), \tag{3.30}$$

so the relation is transitive. Therefore, $\equiv_{P,i}^{\pi,\tau}$ is an equivalence relation. □

Remark 3.1 (Signature Equality Does Not Imply Continuation-Set Equality)**.** Let

$$u = r|_{B_i^{\pi,\tau}} \tag{3.31}$$

and

$$v = r'|_{B_i^{\pi,\tau}} \tag{3.32}$$

for reachable prefix states $r, r' \in \mathcal{R}_i^{\pi}(P)$. In general,

$$u \equiv_{P,i}^{\pi,\tau} v \tag{3.33}$$

implies only

$$\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r'). \tag{3.34}$$

It does not imply

$$E_i^{\pi}(r) = E_i^{\pi}(r'). \tag{3.35}$$

Two different continuation sets may produce the same declared decision value, the same count, the same optimum value, or another identical task-relative certificate. Equality of continuation sets follows only when the declared reconstruction-safe signature distinguishes all continuation sets that occur among reachable prefix states.

Proposition 3.2 (Condition for Exact Continuation-Set Recovery)**.** *Define the reachable continuation family*

$$\mathfrak{E}_i(P,\pi) = \{E_i^\pi(r) : r \in \mathcal{R}_i^\pi(P)\}\,. \tag{3.36}$$

Assume that the reconstruction-safe signature assignment is injective with respect to the reachable continuation family, in the sense that

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r') \tag{3.37}$$

implies

$$E_i^\pi(r) = E_i^\pi(r') \tag{3.38}$$

for all reachable prefix states $r, r' \in \mathcal{R}_i^\pi(P)$. Then, for all reachable interface states

$$u = r|_{B_i^{\pi,\tau}} \tag{3.39}$$

and

$$v = r'|_{B_i^{\pi,\tau}}, \tag{3.40}$$

one has

$$u \equiv_{P,i}^{\pi,\tau} v \quad \Longleftrightarrow \quad E_i^\pi(r) = E_i^\pi(r'). \tag{3.41}$$

Proof. Assume first that

$$E_i^\pi(r) = E_i^\pi(r'). \tag{3.42}$$

Because the reconstruction-safe signature is a declared function of the task-relative future behavior, identical continuation data produce the same declared future signature. Hence,

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r'). \tag{3.43}$$

By the definition of the induced interface signature,

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v). \tag{3.44}$$

Therefore,

$$u \equiv_{P,i}^{\pi,\tau} v. \tag{3.45}$$

Conversely, assume

$$u \equiv_{P,i}^{\pi,\tau} v. \tag{3.46}$$

Then

$$\mathrm{Sig}_{\tau,i}^\pi(r) = \mathrm{Sig}_{\tau,i}^\pi(r'). \tag{3.47}$$

The stated injectivity assumption therefore gives

$$E_i^\pi(r) = E_i^\pi(r'). \tag{3.48}$$

□

For exact enumeration with a signature that preserves the complete enumerated continuation family, the injectivity condition is satisfied. For decision, counting, and many optimization tasks, it generally need not be satisfied.

3.3 Semantic Quotient

Definition 3.4 (Semantic State Space)**.** For a fixed declared task τ, define the semantic state space at stage i by

$$\mathcal{S}_i^\tau(P,\pi;\mathfrak{M}) = \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \big/ \equiv_{P,i}^{\pi,\tau}. \tag{3.49}$$

When the task and reconstruction model are fixed, the shorter notation

$$\mathcal{S}_i(P,\pi) \tag{3.50}$$

may be used.

The quotient map is

$$q_i^{\pi,\tau} : \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \twoheadrightarrow \mathcal{S}_i^\tau(P,\pi;\mathfrak{M}), \tag{3.51}$$

with

$$q_i^{\pi,\tau}(u) = [u]_{\equiv_{P,i}^{\pi,\tau}}. \tag{3.52}$$

Proposition 3.3 (Signature Representation of the Semantic Quotient)**.** *The semantic quotient is canonically bijective to the image of the induced future-signature map:*

$$\mathcal{S}_i^\tau(P,\pi;\mathfrak{M}) \cong \widehat{\sigma}_{P,i}^{\pi,\tau}\left(\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})\right). \tag{3.53}$$

Proof. Define

$$\overline{\sigma}_{P,i}^{\pi,\tau} : \mathcal{S}_i^\tau(P,\pi;\mathfrak{M}) \longrightarrow \widehat{\sigma}_{P,i}^{\pi,\tau}\left(\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})\right) \tag{3.54}$$

by

$$\overline{\sigma}_{P,i}^{\pi,\tau}\left([u]_{\equiv_{P,i}^{\pi,\tau}}\right) = \widehat{\sigma}_{P,i}^{\pi,\tau}(u). \tag{3.55}$$

The map is well defined because all members of one equivalence class have the same induced future signature. It is surjective by construction. If two equivalence classes have the same image, then their representatives have the same induced future signature. By Definition 3.3, the representatives are equivalent and therefore belong to the same class. Hence, the map is injective. Therefore, $\overline{\sigma}_{P,i}^{\pi,\tau}$ is a bijection. □

Lemma 3.1 (Next-Value Invariance on Semantic Classes)**.** *Let*

$$u, v \in \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \tag{3.56}$$

at a nonterminal reconstruction stage $i < n$. If

$$u \equiv_{P,i}^{\pi,\tau} v, \tag{3.57}$$

then

$$\widehat{\Lambda}_i^{\pi,\tau}(u) = \widehat{\Lambda}_i^{\pi,\tau}(v). \tag{3.58}$$

Proof. Choose reachable prefix states

$$r, r' \in \mathcal{R}_i^{\pi}(P) \tag{3.59}$$

such that

$$r|_{B_i^{\pi,\tau}} = u \tag{3.60}$$

and

$$r'|_{B_i^{\pi,\tau}} = v. \tag{3.61}$$

Since

$$u \equiv_{P,i}^{\pi,\tau} v, \tag{3.62}$$

one has

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v). \tag{3.63}$$

By the definition of the induced signature,

$$\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r'). \tag{3.64}$$

Condition S2 of Chapter 2 therefore gives

$$\Lambda_i^{\pi}(r) = \Lambda_i^{\pi}(r'). \tag{3.65}$$

By the definition of the interface-level admissible next-value map,

$$\widehat{\Lambda}_i^{\pi,\tau}(u) = \Lambda_i^{\pi}(r) \tag{3.66}$$

and

$$\widehat{\Lambda}_i^{\pi,\tau}(v) = \Lambda_i^{\pi}(r'). \tag{3.67}$$

Hence,

$$\widehat{\Lambda}_i^{\pi,\tau}(u) = \widehat{\Lambda}_i^{\pi,\tau}(v). \tag{3.68}$$

□

Thus, the set of admissible next values depends only on the semantic equivalence class and not on the selected interface-state representative.

Definition 3.5 (Quotient Transition)**.** Let $i < n$, let

$$[u]_{\equiv_{P,i}^{\pi,\tau}} \in \mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M}), \tag{3.69}$$

and let

$$a \in \widehat{\Lambda}_i^{\pi,\tau}(u). \tag{3.70}$$

Choose any reachable prefix state

$$r \in \mathcal{R}_i^{\pi}(P) \tag{3.71}$$

such that

$$r|_{B_i^{\pi,\tau}} = u. \tag{3.72}$$

Define

$$\overline{\delta}_i^{\pi,\tau}\left([u]_{\equiv_{P,i}^{\pi,\tau}}, a\right) = \left[\delta_i^{\pi}(r,a)\Big|_{B_{i+1}^{\pi,\tau}}\right]_{\equiv_{P,i+1}^{\pi,\tau}}. \tag{3.73}$$

Proposition 3.4 (Well-Definedness of the Quotient Transition)**.** *The quotient* *transition*

$$\overline{\delta}_i^{\pi,\tau}\left([u]_{\equiv_{P,i}^{\pi,\tau}}, a\right) \tag{3.74}$$

is independent of

- *the selected representative u of the semantic class;*
- *the selected reachable prefix state r satisfying $r|_{B_i^{\pi,\tau}} = u$.*

Proof. Let

$$u \equiv_{P,i}^{\pi,\tau} v. \tag{3.75}$$

Choose reachable prefix states

$$r, r' \in \mathcal{R}_i^{\pi}(P) \tag{3.76}$$

such that

$$r|_{B_i^{\pi,\tau}} = u \tag{3.77}$$

and

$$r'|_{B_i^{\pi,\tau}} = v. \tag{3.78}$$

By Lemma 3.1,

$$\widehat{\Lambda}_i^{\pi,\tau}(u) = \widehat{\Lambda}_i^{\pi,\tau}(v). \tag{3.79}$$

Hence, the same next value a is admissible for both representatives. Furthermore,

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v), \tag{3.80}$$

and therefore

$$\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r'). \tag{3.81}$$

Condition S3 of Chapter 2 yields

$$\mathrm{Sig}_{\tau,i+1}^{\pi}\Big(\delta_i^{\pi}(r,a)\Big) = \mathrm{Sig}_{\tau,i+1}^{\pi}\Big(\delta_i^{\pi}(r',a)\Big). \tag{3.82}$$

By interface factorization at stage $i+1$,

$$\widehat{\sigma}_{P,i+1}^{\pi,\tau}\left(\delta_i^{\pi}(r,a)\Big|_{B_{i+1}^{\pi,\tau}}\right) = \widehat{\sigma}_{P,i+1}^{\pi,\tau}\left(\delta_i^{\pi}(r',a)\Big|_{B_{i+1}^{\pi,\tau}}\right). \tag{3.83}$$

Thus,

$$\delta_i^{\pi}(r,a)\Big|_{B_{i+1}^{\pi,\tau}} \equiv_{P,i+1}^{\pi,\tau} \delta_i^{\pi}(r',a)\Big|_{B_{i+1}^{\pi,\tau}}. \tag{3.84}$$

Therefore, both choices determine the same semantic successor class. □

The quotient transition is therefore a well-defined partial stage-to-stage map on semantic-state and admissible-value pairs:

$$\overline{\delta}_i^{\pi,\tau} : \Big\{([u],a) : [u] \in \mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}),\ a \in \widehat{\Lambda}_i^{\pi,\tau}(u)\Big\} \longrightarrow \mathcal{S}_{i+1}^{\tau}(P,\pi;\mathfrak{M}). \tag{3.85}$$

Definition 3.6 (Maximum Semantic-State Count)**.** For a fixed declared task τ, define

$$S_{\mathrm{CPR}}(P,\pi;\tau,\mathfrak{M}) = \max_{0\le i\le n} \left|\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M})\right|. \tag{3.86}$$

When the reconstruction model is fixed, the shorter notation

$$S_{\mathrm{CPR}}(P,\pi;\tau) \tag{3.87}$$

may be used. If both the task and reconstruction model are fixed, one may write

$$S_{\mathrm{CPR}}(P, \pi). \tag{3.88}$$

At every reconstruction stage,

$$|\mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M})| \le \left|\mathcal{U}_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})\right| \le |\mathcal{U}_i(P, \pi, \tau; \mathfrak{M})|\,. \tag{3.89}$$

3.4 Width–State Bound

Theorem 3.1 (Width–State Bound)**.** *For every finite problem instance P, every admissible reconstruction ordering π, every fixed declared task τ, and every fixed reconstruction model $\mathfrak{M}$,*

$$S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \le q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}. \tag{3.90}$$

Proof. For every reconstruction stage i,

$$|\mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M})| \le \left|\mathcal{U}_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M})\right|. \tag{3.91}$$

Moreover,

$$\mathcal{U}_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M}) \subseteq \mathcal{U}_i(P, \pi, \tau; \mathfrak{M}), \tag{3.92}$$

where

$$\mathcal{U}_i(P, \pi, \tau; \mathfrak{M}) = \prod_{v \in B_i^{\pi,\tau}} D_v. \tag{3.93}$$

Therefore,

$$\begin{aligned} |\mathcal{U}_i(P, \pi, \tau; \mathfrak{M})| &= \prod_{v \in B_i^{\pi,\tau}} |D_v| && (3.94)\\ &\le \prod_{v \in B_i^{\pi,\tau}} q(P) && (3.95)\\ &= q(P)^{|B_i^{\pi,\tau}|}. && (3.96) \end{aligned}$$

By the Interface Bound of Chapter 2,

$$|B_i^{\pi,\tau}| \le \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1. \tag{3.97}$$

Hence,

$$|\mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M})| \le q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}. \tag{3.98}$$

Taking the maximum over all stages i yields

$$S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \le q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}. \tag{3.99}$$

□

The theorem uses only that the semantic state space is a quotient of the reachable interface-state space and that the latter is contained in the raw assignment space of the selected minimum active interface. It does not require equality of feasible continuation sets inside one semantic class.

3.5 Binary Semantic-State Dimension

Definition 3.7 (Binary Semantic-State Dimension)**.** For a fixed declared task τ and reconstruction model $\mathfrak{M}$, define

$$D_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) = \lceil \log_2 (\max \{1,\ S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M})\}) \rceil . \tag{3.100}$$

If $q(P) \geq 2$, the Width–State Bound implies

$$D_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \leq \lceil (\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1) \log_2 q(P) \rceil . \tag{3.101}$$

If $q(P) = 1$, every local domain is a singleton. Consequently, every raw interface-state space contains exactly one assignment, and therefore

$$S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) = 1 \tag{3.102}$$

and

$$D_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) = 0. \tag{3.103}$$

The binary semantic-state dimension is the information-theoretic number of bits required to index one semantic class among the maximum number of classes occurring at any reconstruction stage. It does not imply that the semantic quotient can be constructed, represented, or traversed within polynomial time.

3.6 Task Dependence and Semantic Safety

The semantic equivalence relation depends on the declared computational task because the reconstruction-safe future-signature family is task-relative. For decision, the signature must preserve the declared decision value together with the admissible next-value and successor structure required by S2 and S3. For counting, it must additionally preserve the required continuation count. For enumeration, it must preserve the complete declared output information. For optimization, it must preserve the declared optimum value, optimal witness information, or another explicitly specified optimization certificate. Consequently, the relations

$$\equiv_{P,i}^{\pi,\mathrm{dec}}, \qquad \equiv_{P,i}^{\pi,\mathrm{count}}, \qquad \equiv_{P,i}^{\pi,\mathrm{enum}}, \qquad \equiv_{P,i}^{\pi,\mathrm{opt}} \tag{3.104}$$

need not coincide. This indicates a possible distinction between the task-relative relations. It does not assert that they differ for every individual problem instance.

Proposition 3.5 (Task-Relative Semantic Safety)**.** *Let*

$$u, v \in \mathcal{U}_i^{\mathrm{reach}}(P, \pi, \tau; \mathfrak{M}). \tag{3.105}$$

If

$$u \equiv_{P,i}^{\pi,\tau} v, \tag{3.106}$$

then replacing u *by* v *preserves*

- *the declared task-relative future value;*
- *the admissible next-value set;*
- *the semantic class of every successor obtained under the same admissible next value.*

Proof. Choose reachable prefix states

$$r, r' \in \mathcal{R}_i^\pi(P) \tag{3.107}$$

such that

$$r|_{B_i^{\pi,\tau}} = u \tag{3.108}$$

and

$$r'|_{B_i^{\pi,\tau}} = v. \tag{3.109}$$

Since

$$u \equiv_{P,i}^{\pi,\tau} v, \tag{3.110}$$

one has

$$\widehat{\sigma}_{P,i}^{\pi,\tau}(u) = \widehat{\sigma}_{P,i}^{\pi,\tau}(v). \tag{3.111}$$

By interface factorization,

$$\mathrm{Sig}_{\tau,i}^{\pi}(r) = \mathrm{Sig}_{\tau,i}^{\pi}(r'). \tag{3.112}$$

Condition S1 preserves the declared task-relative future value. Condition S2 gives

$$\Lambda_i^\pi(r) = \Lambda_i^\pi(r'), \tag{3.113}$$

and therefore preserves the interface-level admissible next-value set. Finally, for every common admissible next value a, condition S3 gives

$$\mathrm{Sig}_{\tau,i+1}^{\pi}\Big(\delta_i^\pi(r, a)\Big) = \mathrm{Sig}_{\tau,i+1}^{\pi}\Big(\delta_i^\pi(r', a)\Big). \tag{3.114}$$

After projection onto the next minimum interface, the successor states therefore belong to the same semantic equivalence class. □

Remark 3.2 (Scope of Semantic Safety)**.** Task-relative semantic safety does not imply that

$$E_i^\pi(r) = E_i^\pi(r') \tag{3.115}$$

for two prefix states represented by equivalent interface states. It also does not imply that the two states are interchangeable for a different computational task. For exact enumeration with a continuation-set-complete signature, semantic equivalence may imply equality of the enumerated continuation families. For decision, counting, or non-injective optimization signatures, such a conclusion is not generally available.

3.7 Structural Interpretation

For every reconstruction stage i, the raw interface-state space is

$$\mathcal{U}_i(P, \pi, \tau; \mathfrak{M}) = \prod_{v \in B_i^{\pi,\tau}} D_v. \tag{3.116}$$

Its cardinality satisfies

$$|\mathcal{U}_i(P, \pi, \tau; \mathfrak{M})| = \prod_{v \in B_i^{\pi,\tau}} |D_v| \tag{3.117}$$

$$\le q(P)^{|B_i^{\pi,\tau}|} \tag{3.118}$$

$$\leq q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}. \tag{3.119}$$

The reachable interface-state space

$$\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \subseteq \mathcal{U}_i(P,\pi,\tau;\mathfrak{M}) \tag{3.120}$$

removes interface assignments that cannot arise from any reachable prefix state under the declared reconstruction model. The induced future-signature map

$$\widehat{\sigma}_{P,i}^{\pi,\tau} : \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \longrightarrow \Sigma_{\tau,i}^{\pi} \tag{3.121}$$

assigns to every reachable interface state its reconstruction-safe, task-relative future description. The semantic state space is the quotient

$$\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}) = \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \Big/ {\equiv_{P,i}^{\pi,\tau}}. \tag{3.122}$$

Equivalently,

$$\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}) \cong \widehat{\sigma}_{P,i}^{\pi,\tau}\left(\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})\right). \tag{3.123}$$

There is a surjective quotient map

$$q_i^{\pi,\tau} : \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \twoheadrightarrow \mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}). \tag{3.124}$$

Consequently,

$$|\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M})| \leq \left|\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})\right| \leq |\mathcal{U}_i(P,\pi,\tau;\mathfrak{M})|. \tag{3.125}$$

The complete semantic construction is therefore

$$\mathcal{R}_i^{\pi}(P) \xrightarrow{\rho_i^{\pi,\tau}} \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \xrightarrow{\widehat{\sigma}_{P,i}^{\pi,\tau}} \widehat{\sigma}_{P,i}^{\pi,\tau}\left(\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})\right) \cong \mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}). \tag{3.126}$$

At the global level, the structural relation is

$$\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M}) \longrightarrow S_{\mathrm{CPR}}(P,\pi;\tau,\mathfrak{M}). \tag{3.127}$$

The arrow denotes an upper-bound control relationship established by the Width–State Bound. It does not denote a causal or algorithmic dependency. The width controls the maximum number of processed components that must remain simultaneously active, subject to the adopted normalization. The semantic-state count records the maximum number of reconstruction-safe task-signature classes encountered at any stage. Neither quantity alone establishes a polynomial runtime. A complete runtime analysis must additionally account for the costs of

- constructing the declared reconstruction model;
- generating reachable interface states;
- computing semantic signatures and quotient classes;
- generating and processing semantic transitions;
- reconstructing the declared output;
- verifying correctness;
- and materializing the required result.

The following chapter develops this complete operation-count framework.

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 this chapter, 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 therefore relative to

$$(P, \pi, \tau; \mathfrak{M}).$$

When no ambiguity can arise, the dependence on τ and $\mathfrak{M}$ may be suppressed. 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

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

The six terms are defined as disjoint accounting categories. Every elementary operation of the declared CPR procedure must be assigned to exactly one of these categories. No elementary operation may be counted simultaneously in two different cost components. 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 to both;
- writing an output object belongs to N_{Output}, not again to $N_{\text{Reconstruct}}$;
- verification steps belong to N_{Verify}, unless correctness follows directly from a proved construction and no separate verification operation is executed.

The accounting convention must be fixed before any count-level comparison is performed.

4.2 Build Cost

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

Depending on the declared model, the build cost may include

- parsing and validating the encoded problem instance;
- constructing the finite component representation $V(P)$;
- constructing the local domains D_v;
- constructing or receiving the admissible ordering π;
- evaluating the admissibility predicate $\mathrm{Adm}_{P,\tau}(\pi)$;
- initializing reachable-prefix representations;
- constructing auxiliary indices, incidence structures, or constraint tables;
- and allocating the data structures required by the reconstruction process.

The build cost does not include later semantic quotienting or stage-to-stage state 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.

4.3 Reduction Cost

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

This cost may include

- determining task-sufficient candidate interfaces;
- testing or certifying interface sufficiency according to the declared model;
- selecting cardinality-minimal interfaces when this is performed algorithmically;
- computing the values $b_i^{\pi,\tau}(P)$;
- projecting reachable prefix states onto active interfaces;
- constructing induced interface signatures;
- forming semantic equivalence classes;
- canonicalizing semantic-state representatives;
- and constructing static quotient data used in later transitions.

The reduction cost must include the work required to construct the semantic quotient. The semantic quotient construction includes the computation of the equivalence relation induced by

$$u \equiv^{\tau}_{P,i} v$$

and the resulting quotient representation

$$\mathcal{S}^{\tau}_{i}(P, \pi; \mathfrak{M}).$$

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 during all stages, 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. The distinction between reduction and transition cost is determined by the temporal role of the operation. Operations that construct semantic objects before reconstruction starts, including the computation of induced signatures, equivalence classes, and quotient representatives, belong to N_{Reduce}. Operations that evaluate, update, or traverse already constructed semantic states during the reconstruction process belong to $N_{\text{Transition}}$. Therefore, a precomputed semantic quotient is counted once in N_{Reduce}, whereas repeated semantic comparisons or successor-state computations performed during execution are counted in $N_{\text{Transition}}$. No operation contributing to the same semantic decision may be counted in both categories. In the notation of the preceding chapters, this cost accounts for the construction of the task-sufficient active interface $B^{\pi,\tau}_{i}$ of Chapter 2, the induced reconstruction-safe future signature

$$\widehat{\sigma}^{\pi,\tau}_{P,i}$$

of Chapter 3, and the semantic quotient

$$\mathcal{S}^{\tau}_{i}(P, \pi; \mathfrak{M})$$

of Chapter 3. The reduction cost accounts for realizing, at every reconstruction stage i, the construction

$$B^{\pi,\tau}_{i} \longrightarrow \widehat{\sigma}^{\pi,\tau}_{P,i} \longrightarrow \mathcal{S}^{\tau}_{i}(P, \pi; \mathfrak{M}). \tag{4.2}$$

4.4 Reconstruction Stages and Branching

The reconstruction ordering processes the component set $V(P)$. For the component-by-component reconstruction model introduced in Chapter 2, the number of nonterminal transition stages is

$$m_{\text{stage}}(P, \pi) = |V(P)|. \tag{4.3}$$

The symbol m_{stage} is used instead of $b(P, \pi)$ in order to avoid conflict with the minimum interface size $b^{\pi,\tau}_{i}(P)$. For a reachable interface state

$$u \in \mathcal{U}^{\text{reach}}_{i}(P, \pi, \tau; \mathfrak{M}),$$

let $\widehat{\Lambda}_i^{\pi,\tau}(u)$ denote the well-defined admissible next-value set introduced in Chapter 2. Define the stage-wise branching number by

$$\lambda_i(P,\pi,\tau;\mathfrak{M}) = \begin{cases} \max\limits_{u\in\mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M})} \left|\widehat{\Lambda}_i^{\pi,\tau}(u)\right|, & \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) \neq \varnothing, \\ 0, & \mathcal{U}_i^{\mathrm{reach}}(P,\pi,\tau;\mathfrak{M}) = \varnothing. \end{cases} \tag{4.4}$$

The maximum branching number is defined by

$$\lambda_{\max}(P,\pi,\tau;\mathfrak{M}) = \begin{cases} \max\limits_{0\le i<m_{\mathrm{stage}}(P,\pi)} \lambda_i(P,\pi,\tau;\mathfrak{M}), & m_{\mathrm{stage}}(P,\pi) > 0, \\ 0, & m_{\mathrm{stage}}(P,\pi) = 0. \end{cases} \tag{4.5}$$

Thus, both branching quantities remain well defined when a reachable state set is empty or when the component representation itself is empty. If every stage is deterministic, then

$$\lambda_{\max}(P,\pi,\tau;\mathfrak{M}) \le 1. \tag{4.6}$$

4.5 Semantic Transition Cost

Once the semantic quotient has been constructed, the subsequent CPR execution operates on semantic states rather than on raw interface assignments. A transition is an admissible move from one semantic state at stage i to one semantic state at stage $i+1$, that is, from an element of

$$\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M})$$

to an element of

$$\mathcal{S}_{i+1}^{\tau}(P,\pi;\mathfrak{M}).$$

The quotient transition $\overline{\delta}_i^{\pi,\tau}$ was proved well defined in Chapter 3 under the successor-congruence requirement on the induced future-signature family and realizes the stage-to-stage map

$$\mathcal{S}_i^{\tau}(P,\pi;\mathfrak{M}) \longrightarrow \mathcal{S}_{i+1}^{\tau}(P,\pi;\mathfrak{M}). \tag{4.7}$$

Definition 4.4 (Per-Edge Transition Cost)**.** Let $c_{\mathrm{edge}}(P,\pi,\tau;\mathfrak{M})$ be an upper bound on the elementary-operation cost of processing one admissible semantic transition edge.

This cost may include

- generating one successor candidate;
- testing local admissibility;
- evaluating or retrieving its induced signature;
- locating or constructing the successor semantic class;
- merging duplicate successor representatives;
- and updating transition data.

Whether the construction of a successor's induced signature and semantic class is charged to $N_{\mathrm{Transition}}$ or has already been charged to N_{Reduce} is governed by the accounting convention declared in Section 4.3. If the semantic quotient is precomputed once for every stage, these operations reduce to lookups charged to $N_{\mathrm{Transition}}$, while their construction cost has already been counted in N_{Reduce}. If classes are instead constructed on demand while processing transitions, their construction cost is charged to $N_{\mathrm{Transition}}$ and must not also be counted in N_{Reduce}. Exactly one of the two conventions applies for a given declared model.

Definition 4.5 (Local State-Processing Cost)**.** Let $c_{\mathrm{local}}(P, \pi, \tau; \mathfrak{M})$ be an upper bound on the state-local cost incurred once for one semantic state during one reconstruction stage, independently of the number of outgoing transition edges.

This cost may include state initialization, local bookkeeping, retrieval of cached data, state-level canonicalization, or other work performed once per state.

Definition 4.6 (Per-State Transition Cost)**.** Let $c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M})$ 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

$$\begin{aligned} c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M}) \le \lambda_{\max}(P, \pi, \tau; \mathfrak{M})\, &c_{\mathrm{edge}}(P, \pi, \tau; \mathfrak{M}) \\ &+ c_{\mathrm{local}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.8}$$

The manuscript may use either the per-edge convention or the per-state convention, but the selected convention must be stated explicitly.

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

Using the per-state convention,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \le \sum_{i=0}^{m_{\mathrm{stage}}(P,\pi)-1} &|\mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M})| \\ &\cdot c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.9}$$

If $m_{\mathrm{stage}}(P, \pi) = 0$, the sum is interpreted as the empty sum and therefore has value 0. Consequently,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \le m_{\mathrm{stage}}(P, \pi)\, &S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \\ &\cdot c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.10}$$

Using the per-edge convention, including the local cost incurred once per semantic state,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \le m_{\mathrm{stage}}(P, \pi)\, &S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \\ &\cdot \Big[\lambda_{\max}(P, \pi, \tau; \mathfrak{M})\, c_{\mathrm{edge}}(P, \pi, \tau; \mathfrak{M}) \\ &\qquad + c_{\mathrm{local}}(P, \pi, \tau; \mathfrak{M})\Big]. \end{aligned} \tag{4.11}$$

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

4.6 Width-Based Transition Bound

By the raw interface-state bound and the definition of the maximum active interface size,

$$S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \leq q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})}. \tag{4.12}$$

Since Chapter 2 establishes

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \leq \omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) + 1, \tag{4.13}$$

the Width–State Bound of Chapter 3 gives the normalized estimate

$$S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \leq q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1}. \tag{4.14}$$

Consequently, the per-state transition bound becomes

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \leq m_{\mathrm{stage}}(P, \pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \\ \cdot\, c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.15}$$

Using the normalized CPR-Width,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \leq m_{\mathrm{stage}}(P, \pi)\, q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1} \\ \cdot\, c_{\mathrm{state}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.16}$$

Under the per-edge convention,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \leq m_{\mathrm{stage}}(P, \pi)\, q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \\ \cdot \Big[\lambda_{\max}(P, \pi, \tau; \mathfrak{M})\, c_{\mathrm{edge}}(P, \pi, \tau; \mathfrak{M}) \\ + c_{\mathrm{local}}(P, \pi, \tau; \mathfrak{M})\Big]. \end{aligned} \tag{4.17}$$

Equivalently, using the normalized width,

$$\begin{aligned} N_{\mathrm{Transition}}(P, \pi, \tau; \mathfrak{M}) \leq m_{\mathrm{stage}}(P, \pi)\, q(P)^{\omega_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})+1} \\ \cdot \Big[\lambda_{\max}(P, \pi, \tau; \mathfrak{M})\, c_{\mathrm{edge}}(P, \pi, \tau; \mathfrak{M}) \\ + c_{\mathrm{local}}(P, \pi, \tau; \mathfrak{M})\Big]. \end{aligned} \tag{4.18}$$

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.

4.7 Reconstruction 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, reconstruction may require only an acceptance value or one witness. For counting, it may require the propagation and recovery of exact multiplicities. For enumeration, it may require the explicit recovery of every required solution. For optimization, it may require an optimum value, an optimal witness, all optimal witnesses, or another declared certificate. The reconstruction cost includes operations used to trace predecessor information, expand semantic representatives, recover component values, or assemble complete outputs. It does not include the physical writing of the final outputs; that work belongs to N_{Output}. In the notation of the preceding chapters, reconstruction begins at the terminal semantic quotient

$$\mathcal{S}^{\tau}_{m_{\text{stage}}(P,\pi)}(P, \pi; \mathfrak{M})$$

reached after the last transition stage and recovers from it the declared output required by τ. Materializing that output is accounted for separately by N_{Output}.

4.8 Verification Cost

Definition 4.9 (Verification Cost)**.** The verification cost $N_{\text{Verify}}(P, \pi, \tau; \mathfrak{M})$ 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.

Verification may be omitted only when correctness follows directly from the construction and this implication has been proved. A statement that an output is valid is not sufficient to eliminate the verification term unless no separate verification procedure is actually executed.

4.9 Output Cost

Definition 4.10 (Output Cost)**.** The output cost $N_{\text{Output}}(P, \pi, \tau; \mathfrak{M})$ 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 an enumeration task, the output may be exponentially large in the input length. Let $L_{\text{Output}}(P, \pi, \tau; \mathfrak{M})$ denote the total encoded output length. Assume an elementary output model in which writing one output symbol or bit requires at least a fixed positive number $c_{\text{out}} > 0$ of elementary operations. Then

$$N_{\text{Output}}(P, \pi, \tau; \mathfrak{M}) \geq c_{\text{out}}\, L_{\text{Output}}(P, \pi, \tau; \mathfrak{M}). \tag{4.19}$$

No runtime statement may ignore this unavoidable output-size lower bound.

4.10 Post-Transition Cost

For compact notation, define the post-transition cost by

$$\begin{aligned} R_{\text{post}}(P, \pi, \tau; \mathfrak{M}) = N_{\text{Reconstruct}}&(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Verify}}(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Output}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.20}$$

The complete cost model may then be written as

$$\begin{aligned} N_{\text{CPR}}(P, \pi, \tau; \mathfrak{M}) = N_{\text{Build}}&(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Reduce}}(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Transition}}(P, \pi, \tau; \mathfrak{M}) \\ &+ R_{\text{post}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.21}$$

This is only a notational abbreviation. The six-component decomposition in Definition 4.1 remains the primary complete accounting model.

4.11 Complete Width-Based Runtime Bound

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

$$\begin{aligned} N_{\text{CPR}}(P, \pi, \tau; \mathfrak{M}) \le N_{\text{Build}}&(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Reduce}}(P, \pi, \tau; \mathfrak{M}) \\ &+ m_{\text{stage}}(P, \pi)\, q(P)^{\widehat{\omega}_{\text{rec}}(P,\pi,\tau;\mathfrak{M})} \\ &\qquad \cdot c_{\text{state}}(P, \pi, \tau; \mathfrak{M}) \\ &+ R_{\text{post}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.22}$$

Consequently,

$$\begin{aligned} N_{\text{CPR}}(P, \pi, \tau; \mathfrak{M}) \le N_{\text{Build}}&(P, \pi, \tau; \mathfrak{M}) \\ &+ N_{\text{Reduce}}(P, \pi, \tau; \mathfrak{M}) \\ &+ m_{\text{stage}}(P, \pi)\, q(P)^{\omega_{\text{rec}}(P,\pi,\tau;\mathfrak{M})+1} \\ &\qquad \cdot c_{\text{state}}(P, \pi, \tau; \mathfrak{M}) \\ &+ R_{\text{post}}(P, \pi, \tau; \mathfrak{M}). \end{aligned} \tag{4.23}$$

Proof. By the complete six-component decomposition in Equation (4.1) and the definition of the post-transition cost in Equation (4.20), one obtains

$$N_{\text{CPR}} = N_{\text{Build}} + N_{\text{Reduce}} + N_{\text{Transition}} + R_{\text{post}}. \tag{4.24}$$

The per-state transition bound gives

$$N_{\text{Transition}} \le m_{\text{stage}}\, S_{\text{CPR}}\, c_{\text{state}}. \tag{4.25}$$

By the raw interface-state bound,

$$S_{\text{CPR}} \le q(P)^{\widehat{\omega}_{\text{rec}}}. \tag{4.26}$$

Substitution yields

$$N_{\text{Transition}} \le m_{\text{stage}}\, q(P)^{\widehat{\omega}_{\text{rec}}}\, c_{\text{state}}. \tag{4.27}$$

Substituting this inequality into Equation (4.21) proves Equation (4.22). Chapter 2 established

$$q(P) \ge 1 \tag{4.28}$$

and

$$\widehat{\omega}_{\text{rec}} \le \omega_{\text{rec}} + 1. \tag{4.29}$$

Since the function $x \mapsto q(P)^x$ is nondecreasing for $q(P) \ge 1$, it follows that

$$q(P)^{\widehat{\omega}_{\text{rec}}} \le q(P)^{\omega_{\text{rec}}+1}. \tag{4.30}$$

Therefore, Equation (4.23) follows. □

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

$$\begin{aligned} N_{\text{CPR}}(P,\pi,\tau;\mathfrak{M}) \le{}& N_{\text{Build}}(P,\pi,\tau;\mathfrak{M}) \\ &+ N_{\text{Reduce}}(P,\pi,\tau;\mathfrak{M}) \\ &+ m_{\text{stage}}(P,\pi)\, q(P)^{\widehat{\omega}_{\text{rec}}(P,\pi,\tau;\mathfrak{M})} \\ &\quad \cdot \Big[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\text{edge}}(P,\pi,\tau;\mathfrak{M}) \\ &\qquad + c_{\text{local}}(P,\pi,\tau;\mathfrak{M})\Big] \\ &+ R_{\text{post}}(P,\pi,\tau;\mathfrak{M}). \end{aligned} \tag{4.31}$$

Proof. At every reconstruction stage, at most

$$S_{\text{CPR}}(P,\pi;\tau,\mathfrak{M})$$

semantic states are processed. For each semantic state, the state-local work costs at most

$$c_{\text{local}}(P,\pi,\tau;\mathfrak{M}).$$

Each state has at most

$$\lambda_{\max}(P,\pi,\tau;\mathfrak{M})$$

admissible outgoing transitions, and every transition edge costs at most

$$c_{\text{edge}}(P,\pi,\tau;\mathfrak{M}).$$

Therefore,

$$\begin{aligned} N_{\text{Transition}}(P,\pi,\tau;\mathfrak{M}) \le{}& m_{\text{stage}}(P,\pi)\, S_{\text{CPR}}(P,\pi;\tau,\mathfrak{M}) \\ &\cdot \Big[\lambda_{\max}(P,\pi,\tau;\mathfrak{M})\, c_{\text{edge}}(P,\pi,\tau;\mathfrak{M}) \\ &\qquad + c_{\text{local}}(P,\pi,\tau;\mathfrak{M})\Big]. \end{aligned} \tag{4.32}$$

Using

$$S_{\text{CPR}}(P,\pi;\tau,\mathfrak{M}) \le q(P)^{\widehat{\omega}_{\text{rec}}(P,\pi,\tau;\mathfrak{M})} \tag{4.33}$$

and substituting the resulting transition bound into Equation (4.21) proves Equation (4.31). □

4.12 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 $q_0 > 1$ such that*

$$1 \leq q(P) \leq q_0. \tag{4.34}$$

The lower bound $q(P) \geq 1$ includes deterministic domains, where the state-space contribution may collapse to a constant factor. The upper bound q_0 controls the maximal local domain size. Assume further that

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = O(\log \ell(P)). \tag{4.35}$$

Then there exists a constant $c > 0$ such that

$$q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} \leq \ell(P)^c \tag{4.36}$$

for all sufficiently large inputs. Equivalently,

$$q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} = \ell(P)^{O(1)}. \tag{4.37}$$

Proof. Since $q(P) \leq q_0$ and

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) = O(\log \ell(P)),$$

there exists a constant $C > 0$ such that

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \leq C \log \ell(P)$$

for all sufficiently large instances. Therefore,

$$\begin{aligned} q(P)^{\widehat{\omega}_{\mathrm{rec}}(P,\pi,\tau;\mathfrak{M})} &\leq q_0^{C \log \ell(P)} && (4.38) \\ &= \ell(P)^{C \log q_0}. && (4.39) \end{aligned}$$

Since q_0 and C are constants, the exponent $C \log q_0$ is constant. Hence, the right-hand side is polynomially bounded in $\ell(P)$. □

The corollary controls only the width-dependent state factor. A complete polynomial-time result additionally requires polynomials

$$p_{\mathrm{Build}}, \quad p_{\mathrm{Reduce}}, \quad p_{\mathrm{Transition}}, \quad p_{\mathrm{Reconstruct}}, \quad p_{\mathrm{Verify}}, \quad p_{\mathrm{Output}}$$

such that

$$N_j(P, \pi, \tau; \mathfrak{M}) \leq p_j(\ell(P)) \tag{4.40}$$

for every cost component

$$j \in \{\mathrm{Build}, \mathrm{Reduce}, \mathrm{Transition}, \mathrm{Reconstruct}, \mathrm{Verify}, \mathrm{Output}\}.$$

In particular, one must also control $m_{\text{stage}}(P,\pi)$, $\lambda_{\max}(P,\pi,\tau;\mathfrak{M})$, and either $c_{\text{state}}(P,\pi,\tau;\mathfrak{M})$ or $c_{\text{edge}}(P,\pi,\tau;\mathfrak{M})$ together with $c_{\text{local}}(P,\pi,\tau;\mathfrak{M})$, according to the selected transition-cost convention.

4.13 Sufficient Complete Polynomial-Cost Condition

Let

$$\mathcal{F} = \{P_\alpha : \alpha \in A\} \tag{4.41}$$

be an encoded family of finite problem instances. Assume that the declared task τ, the reconstruction model $\mathfrak{M}$, and the procedure for selecting or constructing an admissible ordering π_P are fixed uniformly over $\mathcal{F}$.

Theorem 4.3 (Sufficient Complete Polynomial-Cost Condition)**.** *Assume that there exist fixed polynomials*

$$p_{\text{Build}},\quad p_{\text{Reduce}},\quad p_{\text{Transition}},\quad p_{\text{Reconstruct}},\quad p_{\text{Verify}},\quad p_{\text{Output}}$$

independent of the individual instance P, such that

$$N_j(P,\pi_P,\tau;\mathfrak{M}) \le p_j(\ell(P)) \tag{4.42}$$

for every $P \in \mathcal{F}$ and every one of the six cost components. Then there exists a polynomial p_{CPR}, independent of P, such that

$$N_{\text{CPR}}(P,\pi_P,\tau;\mathfrak{M}) \le p_{\text{CPR}}(\ell(P)) \tag{4.43}$$

for every $P \in \mathcal{F}$.

Proof. By Equation (4.1),

$$\begin{aligned} N_{\text{CPR}} = {} & N_{\text{Build}} + N_{\text{Reduce}} + N_{\text{Transition}} \\ & + N_{\text{Reconstruct}} + N_{\text{Verify}} + N_{\text{Output}}. \end{aligned} \tag{4.44}$$

Using the assumed uniform bounds gives

$$\begin{aligned} N_{\text{CPR}} \le {} & p_{\text{Build}}(\ell(P)) + p_{\text{Reduce}}(\ell(P)) \\ & + p_{\text{Transition}}(\ell(P)) + p_{\text{Reconstruct}}(\ell(P)) \\ & + p_{\text{Verify}}(\ell(P)) + p_{\text{Output}}(\ell(P)). \end{aligned} \tag{4.45}$$

A finite sum of fixed polynomials is a polynomial. Therefore, the polynomial

$$\begin{aligned} p_{\text{CPR}} = {} & p_{\text{Build}} + p_{\text{Reduce}} + p_{\text{Transition}} \\ & + p_{\text{Reconstruct}} + p_{\text{Verify}} + p_{\text{Output}} \end{aligned} \tag{4.46}$$

satisfies Equation (4.43) uniformly for every $P \in \mathcal{F}$. □

4.14 Classical Baseline and Count-Level Gain

Let

$$N_{\text{Classic}}(P,\tau;\mathcal{A}_{\text{Classic}})$$

denote the complete elementary-operation count of a declared reference algorithm $\mathcal{A}_{\text{Classic}}$ 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 that

$$N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) > 0. \tag{4.47}$$

Definition 4.11 (Count-Level Gain)**.** The count-level gain is an elementary-operation-count metric rather than a measured execution-time metric. The count-level gain of the CPR model relative to the declared classical baseline is

$$G_{\mathrm{count}}(P, \pi, \tau; \mathfrak{M}, \mathcal{A}_{\mathrm{Classic}}) = \frac{N_{\mathrm{Classic}}(P, \tau; \mathcal{A}_{\mathrm{Classic}})}{N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})}. \tag{4.48}$$

A favorable count-level regime exists when

$$N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}) < N_{\mathrm{Classic}}(P, \tau; \mathcal{A}_{\mathrm{Classic}}), \tag{4.49}$$

equivalently,

$$G_{\mathrm{count}}(P, \pi, \tau; \mathfrak{M}, \mathcal{A}_{\mathrm{Classic}}) > 1. \tag{4.50}$$

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 it is not the only admissible classical baseline.

4.15 Measured Wall-Clock Gain

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

$$T_{\mathrm{CPR}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}})$$

denote the measured wall-clock time of a declared CPR implementation $\mathcal{I}_{\mathrm{CPR}}$. Assume that

$$T_{\mathrm{CPR}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}}) > 0. \tag{4.51}$$

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

$$G_{\mathrm{time}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}}, \mathcal{I}_{\mathrm{Classic}}) = \frac{T_{\mathrm{Classic}}(P, \tau; \mathcal{I}_{\mathrm{Classic}})}{T_{\mathrm{CPR}}(P, \pi, \tau; \mathcal{I}_{\mathrm{CPR}})}. \tag{4.52}$$

A count-level gain does not imply a wall-clock gain:

$$G_{\mathrm{count}} > 1 \not\Longrightarrow G_{\mathrm{time}} > 1. \tag{4.53}$$

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.

Therefore, operation-count claims and wall-clock claims must remain separate.

4.16 Structural Runtime Chain

The theory developed in Chapters 2–3 yields the central structural runtime chain

$$\boldsymbol{\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \longrightarrow S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \longrightarrow N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})}. \tag{4.54}$$

Using the normalized width, this may also be written as

$$\boldsymbol{\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \longrightarrow S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M}) \longrightarrow N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M})}. \tag{4.55}$$

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.

4.17 Runtime Non-Implications

Proposition 4.1 (Width Does Not Determine Runtime Alone)**.** *The inequality*

$$\omega_{\mathrm{rec}}(P, \pi_1, \tau; \mathfrak{M}) < \omega_{\mathrm{rec}}(P, \pi_2, \tau; \mathfrak{M}) \tag{4.56}$$

does not by itself imply

$$N_{\mathrm{CPR}}(P, \pi_1, \tau; \mathfrak{M}) < N_{\mathrm{CPR}}(P, \pi_2, \tau; \mathfrak{M}). \tag{4.57}$$

It also does not imply

$$T_{\mathrm{CPR}}(P, \pi_1, \tau; \mathcal{I}_{\mathrm{CPR}}) < T_{\mathrm{CPR}}(P, \pi_2, \tau; \mathcal{I}_{\mathrm{CPR}}), \tag{4.58}$$

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;
- more costly semantic signatures;

- larger branching;
- more expensive quotient transitions;
- greater reconstruction overhead;
- greater verification cost;
- or less favorable memory behavior.

Therefore, width alone does not determine either the complete operation count or the measured wall-clock time. □

Proposition 4.2 (Semantic-State Count Does Not Determine Runtime Alone)**.** *The inequality*

$$S_{\mathrm{CPR}}(P,\pi_1;\tau,\mathfrak{M}) < S_{\mathrm{CPR}}(P,\pi_2;\tau,\mathfrak{M}) \tag{4.59}$$

does not by itself imply

$$N_{\mathrm{CPR}}(P,\pi_1,\tau;\mathfrak{M}) < N_{\mathrm{CPR}}(P,\pi_2,\tau;\mathfrak{M}). \tag{4.60}$$

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

4.18 Conclusion

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

$$\begin{aligned} N_{\mathrm{CPR}} = N_{\mathrm{Build}} &+ N_{\mathrm{Reduce}} + N_{\mathrm{Transition}} \\ &+ N_{\mathrm{Reconstruct}} + N_{\mathrm{Verify}} + N_{\mathrm{Output}}. \end{aligned} \tag{4.61}$$

The six cost components are disjoint accounting categories. Every elementary operation must be assigned to exactly one category. 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,

$$N_{\mathrm{Transition}} \le m_{\mathrm{stage}}\, S_{\mathrm{CPR}}\, \left(\lambda_{\max} c_{\mathrm{edge}} + c_{\mathrm{local}}\right). \tag{4.62}$$

Under the per-state convention,

$$N_{\mathrm{Transition}} \le m_{\mathrm{stage}}\, S_{\mathrm{CPR}}\, c_{\mathrm{state}}. \tag{4.63}$$

The width-based bound controls the semantic-state contribution through

$$S_{\mathrm{CPR}} \le q(P)^{\widehat{\omega}_{\mathrm{rec}}} \le q(P)^{\omega_{\mathrm{rec}}+1}. \tag{4.64}$$

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

$$\boldsymbol{\omega}_{\mathbf{rec}} \longrightarrow \boldsymbol{S}_{\mathbf{CPR}} \longrightarrow \boldsymbol{N}_{\mathbf{CPR}}. \tag{4.65}$$

A refined runtime-accounting chain is

$$\boldsymbol{\omega}_{\mathbf{rec}} \longrightarrow \boldsymbol{S}_{\mathbf{CPR}} \longrightarrow \boldsymbol{N}_{\mathbf{Transition}} \longrightarrow \boldsymbol{N}_{\mathbf{CPR}}. \tag{4.66}$$

The refined chain makes explicit that the semantic-state count enters the complete operation count primarily through transition processing. It does not imply that the remaining cost components are determined by the transition cost or by the reconstruction width. The semantic-state count S_{CPR} and the binary semantic-state dimension D_{CPR} of Chapter 3 provide structural bounds on the number and representation size of semantic states. They do not directly equal the total operation count N_{CPR}. The complete runtime analysis additionally requires the transition, verification, reconstruction, and output costs defined in this chapter. The following chapter formulates the complete boundary conditions and uniform polynomial-cost requirements needed for CPR tractability.

Chapter Summary

The complete CPR pipeline decomposes into six accounted components: construction of the declared representation (N_{Build}), formation of active interfaces and reduced representations (N_{Reduce}), semantic-transition processing ($N_{\mathrm{Transition}}$), reconstruction of task-relevant output ($N_{\mathrm{Reconstruct}}$), correctness verification (N_{Verify}), and output materialization (N_{Output}). Their sum is the complete operation count

$$N_{\mathrm{CPR}}(P, \pi, \tau; \mathfrak{M}).$$

The width-based and edge-based runtime bounds developed above express $N_{\mathrm{Transition}}$ in terms of the semantic-state count $S_{\mathrm{CPR}}(P, \pi; \tau, \mathfrak{M})$, which is itself controlled by the reconstruction width

$$\omega_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M})$$

through the Width–State Bound. The classical baseline count N_{Classic} and the resulting count-level gain G_{count} are declared separately from measured wall-clock performance. Neither reconstruction width, semantic-state control, nor count-level gain alone establishes a measured runtime advantage. The following chapter formalizes the boundary conditions BC1–BC4 under which this operation-count model yields a uniform polynomial-time guarantee.

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 certify that Controlled Partial Reconstruction admits a polynomially bounded execution over an entire uniformly encoded problem family. This chapter formulates those conditions and establishes the resulting tractability criterion.

Throughout this chapter, all elementary-operation counts are interpreted in one fixed deterministic bit-cost computation model. For definiteness, the model may be taken to be a deterministic multi-tape Turing machine. Alternatively, 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 when 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

$$X_{\mathcal{F}} \subseteq \{0,1\}^*$$

be the set of valid encodings of the family, and let

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}.$$

For every $x \in X_{\mathcal{F}}$, let

$$\ell(P_x) = |x| \tag{5.1}$$

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. It must be 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. Finally, every component of the complete runtime model introduced in Chapter 4 must be polynomially bounded.

These requirements are organized through four boundary conditions:

1. BC1: explicit uniform structural representation;
2. BC2: polynomial uniform constructibility;
3. BC3: task-relative soundness and completeness;

4. BC4: polynomial semantic-state and transition-structure control.

The boundary conditions are required components of the declared CPR tractability certificate. They do not, when 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

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}$$

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 $V(P_x)$;
- the local domains D_v;
- the admissibility predicate for reconstruction orderings;
- the representation of an admissible ordering when the ordering is supplied as part of the encoded instance, or one uniform procedure for constructing such an ordering;
- the reachable prefix-state spaces;
- the declared active reconstruction interfaces together with their task-sufficiency certification;
- the reconstruction-safe prefix-state signature family $\mathrm{Sig}^{\pi_x}_{\tau,i}$;
- the reachable interface-state spaces;
- the induced reconstruction-safe future-signature family;
- the semantic equivalence relation and semantic quotient;
- the quotient transition rule;
- the reconstruction operator;
- the verification rule;
- the required output representation;
- and one uniform validity predicate $\mathrm{Valid}_{\mathcal{F}}(x)$ satisfying

$$\mathrm{Valid}_{\mathcal{F}}(x) = 1 \iff x \in X_{\mathcal{F}}.$$

All objects must be specified by one uniform finite description that applies to the complete encoded family.

The reconstruction-safe prefix-state signature

$$\mathrm{Sig}^{\pi_x}_{\tau,i}$$

introduced in Chapter 2 factors through the reachable interface-state representation. The induced reconstruction-safe future signature on reachable interface states is denoted in Chapter 3 by

$$\widehat{\sigma}^{\pi_x,\tau}_{P_x,i}.$$

The semantic equivalence relation and semantic quotient are defined from this induced signature.

BC1 is a representation condition. It requires the mathematical objects on which the CPR analysis is based to be explicitly declared. It does not assert that these objects are inexpensive to construct. It also does not assert 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

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}$$

satisfies BC2 for task τ and reconstruction model $\mathfrak{M}$ if there exists one deterministic preprocessing algorithm

$$\mathcal{A}_{\text{Preprocess}}$$

and one polynomial

$$p_{\text{BC2}} : \mathbb{N} \longrightarrow \mathbb{N}$$

such that, for every $x \in \{0,1\}^*$,

$$N_{\mathcal{A}_{\text{Preprocess}}}(x) \leq p_{\text{BC2}}(|x|), \tag{5.2}$$

where $N_{\mathcal{A}_{\text{Preprocess}}}(x)$ denotes the number of elementary operations executed by the preprocessing algorithm.

The algorithm first decides whether $x \in X_{\mathcal{F}}$. If $x \notin X_{\mathcal{F}}$, it rejects the encoding within the same polynomial bound. If $x \in X_{\mathcal{F}}$, it constructs or reads all static structural data required before dynamic semantic-state processing begins.

The constructed or supplied data include, whenever applicable,

- polynomial-time validation of whether $x \in X_{\mathcal{F}}$;
- 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, indexing, and constraint structures;
- and all other data assigned to N_{Build} or to 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 representation or constructs one uniformly from x. In both cases, the resulting ordering must satisfy

$$\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M}), \tag{5.3}$$

where $\mathcal{A}_\pi$ denotes the ordering-supply component of $\mathcal{A}_{\text{Preprocess}}$.

The operation count of $\mathcal{A}_\pi$ is included in N_{Build}. It is 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 must be assigned exactly once according to the disjoint accounting convention fixed in Chapter 4.

BC2 requires polynomial constructibility of every structural object that the declared model assigns to preprocessing. The precise allocation between N_{Build} and N_{Reduce} must follow the disjoint accounting convention fixed in Chapter 4.

If the declared CPR model constructs semantic classes or quotient transitions dynamically, BC2 does not require their complete precomputation. In that case, BC2 requires only the polynomial construction of the static data needed to perform those dynamic operations, while their actual execution costs are assigned to $N_{\text{Transition}}$.

BC2 is uniform. The same algorithm and the same polynomial bound must apply to every binary input. 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

$$\mathrm{Sol}_\tau(P_x)$$

denote the complete set of feasible objects relevant to that task, whenever such a solution-set representation is applicable.

Decision

For a decision task, let

$$\mathrm{Dec}_\tau(P_x) \in \{0, 1\}$$

denote the correct task-relative decision value, and define

$$\mathrm{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) \in \{0, 1\}.$$

Soundness and completeness require

$$\mathrm{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = \mathrm{Dec}_\tau(P_x). \tag{5.4}$$

For an existential feasibility decision task, this specializes to

$$\mathrm{Sol}_\tau(P_x) \neq \varnothing \quad \Longleftrightarrow \quad \mathrm{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = 1. \tag{5.5}$$

Thus, 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.

Enumeration

For an enumeration task, soundness and completeness require

$$\mathrm{Output}_{\mathrm{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) = \mathrm{Sol}_\tau(P_x). \tag{5.6}$$

Equivalently, soundness requires

$$\mathrm{Output}_{\mathrm{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) \subseteq \mathrm{Sol}_\tau(P_x), \tag{5.7}$$

and completeness requires

$$\mathrm{Sol}_\tau(P_x) \subseteq \mathrm{Output}_{\mathrm{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}). \tag{5.8}$$

Counting

For an exact counting task, soundness and completeness require

$$\mathrm{CPRCount}(P_x, \pi_x, \tau; \mathfrak{M}) = \left|\mathrm{Sol}_\tau(P_x)\right|. \tag{5.9}$$

Optimization

For an optimization task, let

$$\mathrm{OptVal}_\tau(P_x)$$

denote the exact optimum value, and let

$$\mathrm{OptSol}_\tau(P_x)$$

denote the set of all feasible objects attaining that value.

The declared optimization task must specify whether the required output is

- the optimum value;
- one optimal witness;
- all optimal witnesses;
- or another explicitly declared optimality certificate.

The output requirement must be fixed before the CPR model is evaluated.

For an optimum-value task, BC3 requires

$$\mathrm{CPROptVal}(P_x, \pi_x, \tau; \mathfrak{M}) = \mathrm{OptVal}_\tau(P_x). \tag{5.10}$$

For a task requiring one optimal witness, BC3 requires

$$\mathrm{CPROptWitness}(P_x, \pi_x, \tau; \mathfrak{M}) \in \mathrm{OptSol}_\tau(P_x). \tag{5.11}$$

For a task requiring all optimal witnesses, BC3 requires

$$\mathrm{CPROptOutput}(P_x, \pi_x, \tau; \mathfrak{M}) = \mathrm{OptSol}_\tau(P_x). \tag{5.12}$$

No weaker output object may silently replace the declared task requirement.

Task-Evaluation Form

More generally, let

$$\Theta_\tau(P_x)$$

denote the canonically normalized complete output required by the declared task. Let

$$\Theta_\tau^{\mathrm{CPR}}(P_x, \pi_x; \mathfrak{M})$$

denote the correspondingly normalized output produced by the CPR procedure.

Then BC3 requires

$$\Theta_\tau^{\mathrm{CPR}}(P_x, \pi_x; \mathfrak{M}) = \Theta_\tau(P_x). \tag{5.13}$$

When the declared task permits several equally valid outputs, the normalization rule or the corresponding acceptance relation must be specified as part of the task definition.

The decision, enumeration, counting, and optimization conditions above are special cases of the task-relative correctness requirement. BC3 is a correctness condition. It 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

$$N_{\mathrm{sem,state}}(P_x, \pi_x, \tau; \mathfrak{M}) = \sum_{i=0}^{m_{\mathrm{stage}}(P_x,\pi_x)} \left|\mathcal{S}_i^\tau(P_x, \pi_x; \mathfrak{M})\right|. \tag{5.14}$$

This quantity counts stage-indexed semantic states. The same abstract state occurring at two different reconstruction stages is counted twice because 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

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}$$

satisfies BC4 for task τ, reconstruction model $\mathfrak{M}$, and admissible orderings π_x if there exist polynomials

$$p_{\mathrm{SemanticVertex}}, \qquad p_{\mathrm{SemanticEdge}} : \mathbb{N} \longrightarrow \mathbb{N},$$

independent of x, such that for every $x \in X_{\mathcal{F}}$,

$$N_{\mathrm{sem,state}}(P_x, \pi_x, \tau; \mathfrak{M}) \le p_{\mathrm{SemanticVertex}}(|x|) \tag{5.15}$$

and

$$E_{\mathrm{sem}}(P_x, \pi_x, \tau; \mathfrak{M}) \le p_{\mathrm{SemanticEdge}}(|x|). \tag{5.16}$$

BC4 controls semantic-graph cardinalities only. It does not assert that semantic vertices or edges have constant-size representations, nor that one vertex or edge can be generated or processed in constant or polynomial time. Those requirements belong to the component-wise operation-count bounds.

The first inequality controls the total number of stage-indexed reachable semantic-state occurrences. The second controls the complete number of reachable semantic transition edges.

Since

$$S_{\mathrm{CPR}}(P_x, \pi_x; \tau, \mathfrak{M}) = \max_{0 \le i \le m_{\mathrm{stage}}(P_x, \pi_x)} \left| \mathcal{S}_i^{\tau}(P_x, \pi_x; \mathfrak{M}) \right|, \tag{5.17}$$

one also has

$$S_{\mathrm{CPR}}(P_x, \pi_x; \tau, \mathfrak{M}) \le N_{\mathrm{sem,state}}(P_x, \pi_x, \tau; \mathfrak{M}). \tag{5.18}$$

Therefore, BC4 also implies polynomial control of the maximum stage-wise semantic-state count.

The polynomial cardinality of the semantic-state space does not imply that each state can be generated, compared, or encoded in polynomial time unless these operations are included in the corresponding runtime bounds.

A short binary description of one semantic state does not imply that only polynomially many semantic states occur. Similarly, 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 certification condition. It does not by itself imply that the processing of 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.

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

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}$$

be a uniformly encoded family of finite decision-problem instances.

The family is called CPR-tractable if there exist

- one declared decision task τ;

- one uniform CPR reconstruction model $\mathfrak{M}$;
- one uniform deterministic preprocessing algorithm $\mathcal{A}_{\text{Preprocess}}$, containing an ordering-supply component $\mathcal{A}_\pi$, such that

$$\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M});$$

- and six fixed polynomial functions

$$p_j : \mathbb{N} \longrightarrow \mathbb{N}, \qquad j \in J,$$

  where

$$J = \{\text{Build}, \text{Reduce}, \text{Transition}, \text{Reconstruct}, \text{Verify}, \text{Output}\}.$$

The validity predicate is required to be decidable within the preprocessing bound, BC1–BC4 must hold uniformly, and for every $x \in X_{\mathcal{F}}$,

$$N_j(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_j(|x|) \tag{5.19}$$

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

$$L_{\mathcal{F}} = \{x \in X_{\mathcal{F}} : P_x \text{ is a yes-instance for the declared task } \tau\}. \tag{5.20}$$

Every encoding outside $X_{\mathcal{F}}$ is rejected by the uniform validity procedure within the polynomial preprocessing bound specified by BC2.

Definition 5.6 (The Language Class P_{CPR})**.** Define

$$\mathsf{P}_{\text{CPR}} = \Big\{ L \subseteq \{0,1\}^* : L = L_{\mathcal{F}} \text{ for some uniformly encoded} \\ \text{CPR-tractable decision family } \mathcal{F} \Big\}. \tag{5.21}$$

Thus, membership in P_{CPR} 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 the 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.

The class is not intended as a replacement for standard complexity classes, but as a certified subclass induced by the CPR framework.

5.7 Uniform Tractability

Theorem 5.1 (Uniform CPR Tractability)**.** *Let*

$$\mathcal{F} = \{P_x : x \in X_{\mathcal{F}}\}$$

be a uniformly encoded family of finite decision-problem instances, and let $L_{\mathcal{F}}$ be the associated decision language.

Assume that there exist

- *one declared decision task τ;*
- *one uniform CPR reconstruction model $\mathfrak{M}$;*
- *one uniform deterministic preprocessing algorithm $\mathcal{A}_{\text{Preprocess}}$, containing an ordering-supply component $\mathcal{A}_\pi$, and producing*

$$\pi_x = \mathcal{A}_\pi(x) \in \Pi_{\text{adm}}(P_x, \tau; \mathfrak{M});$$

- *and six fixed polynomial functions*

$$p_j : \mathbb{N} \longrightarrow \mathbb{N}, \qquad j \in J.$$

Assume further that the validity predicate is decidable within the BC2 preprocessing bound, that BC1, BC2, BC3, and BC4 hold uniformly, and that, for every $x \in X_{\mathcal{F}}$,

$$N_j(P_x, \pi_x, \tau; \mathfrak{M}) \leq p_j(|x|) \tag{5.22}$$

for every $j \in J$.

Then

$$L_{\mathcal{F}} \in \mathsf{P}. \tag{5.23}$$

Proof. By BC1, the complete CPR decision model and the validity predicate for the encoded family are explicitly and uniformly specified.

By BC2, the validity of every input

$$x \in \{0,1\}^*$$

is decided using a number of elementary operations polynomial in $|x|$.

If $x \notin X_{\mathcal{F}}$, the procedure rejects x. If $x \in X_{\mathcal{F}}$, 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:

$$x \in L_{\mathcal{F}} \quad \Longleftrightarrow \quad \text{CPRDecision}(P_x, \pi_x, \tau; \mathfrak{M}) = 1. \tag{5.24}$$

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,

$$\begin{aligned} N_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) = {} & N_{\text{Build}}(P_x, \pi_x, \tau; \mathfrak{M}) \\ & + N_{\text{Reduce}}(P_x, \pi_x, \tau; \mathfrak{M}) \end{aligned}$$

$$
\begin{aligned}
&+ N_{\text{Transition}}(P_x, \pi_x, \tau; \mathfrak{M}) \\
&+ N_{\text{Reconstruct}}(P_x, \pi_x, \tau; \mathfrak{M}) \\
&+ N_{\text{Verify}}(P_x, \pi_x, \tau; \mathfrak{M}) \\
&+ N_{\text{Output}}(P_x, \pi_x, \tau; \mathfrak{M}). \qquad (5.25)
\end{aligned}
$$

Using the six assumed uniform polynomial bounds gives

$$
\begin{aligned}
N_{\text{CPR}}(P_x, \pi_x, \tau; \mathfrak{M}) \le\ & p_{\text{Build}}(|x|) + p_{\text{Reduce}}(|x|) \\
&+ p_{\text{Transition}}(|x|) + p_{\text{Reconstruct}}(|x|) \\
&+ p_{\text{Verify}}(|x|) + p_{\text{Output}}(|x|). \qquad (5.26)
\end{aligned}
$$

The right-hand side is a finite sum of fixed polynomial functions and is therefore polynomially bounded in $|x|$.

Because the operation counts are interpreted in the fixed deterministic bit-cost computation model declared at the beginning of this chapter, the complete CPR procedure is executable in deterministic polynomial time.

Consequently, membership of every input $x \in \{0,1\}^*$ in $L_{\mathcal{F}}$ is decided by one deterministic polynomial-time procedure.

Hence,

$$L_{\mathcal{F}} \in \mathsf{P}.$$

□

Corollary 5.1 (Certified CPR Languages Are Polynomial-Time Languages)**.** *The framework-dependent language class satisfies*

$$\mathsf{P}_{\text{CPR}} \subseteq \mathsf{P}. \qquad (5.27)$$

Proof. Let

$$L \in \mathsf{P}_{\text{CPR}}.$$

By Definition 5.6, the language L is associated with a uniformly encoded CPR-tractable decision family. The hypotheses of Theorem 5.1 therefore hold.

Hence,

$$L \in \mathsf{P}.$$

□

The inclusion in Equation (5.27) is a consequence of the certification requirements built into the definition of P_{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

$$L^* \in \mathsf{P}_{\mathrm{CPR}}, \tag{5.28}$$

then

$$\mathsf{P} = \mathsf{NP}. \tag{5.29}$$

Proof. By Corollary 5.1,

$$\mathsf{P}_{\mathrm{CPR}} \subseteq \mathsf{P}.$$

Therefore, the hypothesis

$$L^* \in \mathsf{P}_{\mathrm{CPR}}$$

implies

$$L^* \in \mathsf{P}.$$

Since L^* is NP-complete, every language in NP is polynomial-time many-one reducible to L^*. Because $L^* \in \mathsf{P}$, it follows that

$$\mathsf{NP} \subseteq \mathsf{P}.$$

The standard inclusion

$$\mathsf{P} \subseteq \mathsf{NP}$$

also holds.

Consequently,

$$\mathsf{P} = \mathsf{NP}.$$

□

Remark 5.1 (Unproved Premise)**.** The manuscript does not establish the hypothesis

$$L^* \in \mathsf{P}_{\mathrm{CPR}}$$

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.

Therefore, the preceding corollary is a conditional statement only. It does not establish

$$\mathsf{P} = \mathsf{NP}.$$

Chapter Summary

Polynomial CPR tractability requires a uniform certificate over an entire encoded problem family. BC1 requires an explicit uniform reconstruction representation. BC2 requires polynomial uniform construction of all structural data assigned to preprocessing. BC3 requires exact task-relative soundness and completeness. BC4 requires polynomial cardinality bounds for the stage-indexed semantic-state system and its reachable transition graph.

These four structural conditions do not replace the complete runtime model. A CPR decision family is certified as tractable only when all six operation-count components from Chapter 4 are uniformly polynomially bounded.

Consequently,

$$\mathsf{P}_{\mathrm{CPR}} \subseteq \mathsf{P}.$$

If an NP-complete language were proved to belong to $\mathsf{P}_{\mathrm{CPR}}$, then

$$\mathsf{P} = \mathsf{NP}.$$

No such membership result is established in this manuscript. The following chapter examines the complexity-theoretic scope, representative realizations, and limitations of the CPR-WIDTH framework.

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 this chapter, 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 when no ambiguity can arise. A central distinction is required throughout this chapter. For every problem-class realization, the visibly defined structural boundary is first treated as a candidate interface

$$\widetilde{B}_i^{\pi,\tau}.$$

The cardinality-minimal task-sufficient interface defined in Chapter 2 is denoted by

$$B_i^{\pi,\tau}.$$

Unless task sufficiency and cardinality minimality have both been proved, one may conclude only

$$b_i^{\pi,\tau}(P) \leq \left| \widetilde{B}_i^{\pi,\tau} \right|,$$

and therefore

$$\widehat{\omega}_{\mathrm{rec}}(P, \pi, \tau; \mathfrak{M}) \leq \max_i \left| \widetilde{B}_i^{\pi,\tau} \right|.$$

Equality requires an additional cardinality-minimality proof.

6.1 Boolean Satisfiability

Let

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

be a Boolean formula over the variable set

$$V(\Phi) = \{x_1, \ldots, x_n\}.$$

Let

$$\pi = \left(x_{\pi(1)}, \ldots, x_{\pi(n)}\right)$$

be an admissible variable ordering. After reconstruction stage i, the processed and unresolved variable sets are

$$V_i^\pi = \{x_{\pi(1)}, \ldots, x_{\pi(i)}\}$$

and

$$\overline{V_i^\pi} = V(\Phi) \setminus V_i^\pi.$$

A clause $C_j \in \Phi$ is called unresolved at stage i if it contains at least one variable in $\overline{V_i^\pi}$, and resolved at stage i otherwise. Define the structural SAT candidate interface by

$$\widetilde{B}_i^{\Phi,\pi} = \{x \in V_i^\pi : \exists\, C_j \in \Phi \text{ unresolved at stage } i \text{ with } x \text{ occurring in } C_j\}\,. \tag{6.1}$$

This set contains processed variables whose assigned values may still influence residual clauses. It is a natural structural candidate interface. It is not automatically the unique or cardinality-minimal task-sufficient interface. Let

$$r \in \mathcal{R}_i^\pi(\Phi)$$

be a reachable prefix assignment. For the exact satisfying-extension realization, define

$$E_i^{\Phi,\pi}(r) = \left\{ s \in \prod_{x \in \overline{V_i^\pi}} \{0,1\} : r \cup s \models \Phi \right\}. \tag{6.2}$$

Equality of satisfying-extension sets may be used as the reconstruction-safe prefix signature

$$\mathrm{Sig}^\pi_{\mathrm{SAT},i}(r) = E_i^{\Phi,\pi}(r). \tag{6.3}$$

This signature is stronger than the current decision value

$$\mathbf{1}\left[E_i^{\Phi,\pi}(r) \neq \varnothing\right].$$

The decision value therefore induces a coarser task-output partition than the complete extension-set signature. However, the decision value alone is not automatically a reconstruction-safe staged signature because admissible next values and successor-class congruence must still satisfy S2 and S3. The extension-set signature preserves the decision answer, admissible next values, and successor residual classes. Hence, it satisfies the reconstruction-safety conditions S1–S3 of Chapter 2. In the standard variable-assignment realization, processed variables outside $\widetilde{B}_i^{\Phi,\pi}$ occur only in clauses that are already resolved. For reachable prefix assignments, their values therefore do not alter the residual satisfying-extension set. Consequently, the candidate interface determines the signature and

$$b_i^{\pi,\mathrm{SAT}}(\Phi) \leq \left|\widetilde{B}_i^{\Phi,\pi}\right|. \tag{6.4}$$

Thus,

$$\widehat{\omega}_{\mathrm{rec}}(\Phi, \pi, \mathrm{SAT}; \mathfrak{M}) \leq \max_i \left|\widetilde{B}_i^{\Phi,\pi}\right|. \tag{6.5}$$

An equality claim requires a proof that no smaller processed-variable set determines the same reconstruction-safe 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 constraint-satisfaction instance, where

$$V = \{x_1, \dots, x_n\},$$

$$D = (D_{x_1}, \dots, D_{x_n}),$$

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 by

$$\widetilde{B}_i^{P_{\mathrm{CSP}},\pi} = \left\{ x \in V_i^\pi : \begin{array}{l} \exists C \in \mathcal{C} \text{ such that } x \in \mathrm{scope}(C), \\ \mathrm{scope}(C) \cap \overline{V_i^\pi} \neq \varnothing \end{array} \right\}. \tag{6.6}$$

This structural boundary contains processed variables participating in constraints whose scopes still contain unresolved variables. For a reachable prefix assignment r, define the exact residual signature by

$$\mathrm{Sig}^\pi_{\mathrm{CSP},i}(r) = \left\{ s \in \prod_{x \in \overline{V_i^\pi}} D_x : r \cup s \text{ satisfies every constraint in } \mathcal{C} \right\}. \tag{6.7}$$

This extension-set signature is suitable for exact feasibility reconstruction. For counting, optimization, or another declared task, the signature may instead preserve the corresponding normalized task value together with the admissible-next-value and successor information required by S1–S3. Processed variables outside $\widetilde{B}_i^{P_{\mathrm{CSP}},\pi}$ participate only in constraints whose scopes are already resolved. For reachable prefix assignments, they therefore do not influence the remaining compatible assignments. Hence, the candidate interface determines the exact residual signature and

$$b_i^{\pi,\tau}(P_{\mathrm{CSP}}) \le \left| \widetilde{B}_i^{P_{\mathrm{CSP}},\pi} \right|. \tag{6.8}$$

Thus,

$$\widehat{\omega}_{\mathrm{rec}}(P_{\mathrm{CSP}}, \pi, \tau; \mathfrak{M}) \le \max_i \left| \widetilde{B}_i^{P_{\mathrm{CSP}},\pi} \right|. \tag{6.9}$$

The structural boundary is therefore a certified upper bound for this realization. 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 let

$$K = \{1, \dots, k\}$$

be the available color set. For an ordering

$$\pi = (v_{\pi(1)}, \dots, v_{\pi(|V|)}),$$

define the processed and unresolved vertex sets by

$$V_i^\pi = \{v_{\pi(1)}, \dots, v_{\pi(i)}\}$$

and

$$\overline{V_i^\pi} = V \setminus V_i^\pi.$$

Define the structural graph-coloring candidate interface by

$$\widetilde{B}_i^{G,\pi} = \left\{ v \in V_i^\pi : \exists w \in \overline{V_i^\pi} \text{ with } \{v, w\} \in E \right\}. \tag{6.10}$$

This set is exactly the classical vertex-separation boundary of the ordering π. For a reachable proper prefix coloring

$$r : V_i^\pi \longrightarrow K,$$

define the set of proper future extensions by

$$E_i^{G,\pi}(r) = \left\{ s : \overline{V_i^\pi} \to K : r \cup s \text{ is a proper coloring of } G \right\}. \tag{6.11}$$

An exact reconstruction-safe signature is

$$\mathrm{Sig}^\pi_{\mathrm{COL},i}(r) = E_i^{G,\pi}(r). \tag{6.12}$$

The colors assigned to processed vertices outside $\widetilde{B}_i^{G,\pi}$ cannot constrain unresolved vertices because no edge connects those processed vertices to the unresolved suffix. Therefore, the structural boundary determines the extension-set signature. It follows that

$$b_i^{\pi,\tau}(G) \le \left| \widetilde{B}_i^{G,\pi} \right| \tag{6.13}$$

for the declared exact coloring realization. Consequently,

$$\widehat{\omega}_{\mathrm{rec}}(G, \pi, \tau; \mathfrak{M}) \le \max_i \left| \widetilde{B}_i^{G,\pi} \right|. \tag{6.14}$$

The right-hand side is the ordering-specific vertex-separation value. Classical vertex separation therefore provides a structural upper bound for the ordering-specific minimum-interface CPR width. The CPR value may be strictly smaller if semantic redundancy permits a smaller task-sufficient coordinate set or if semantic quotienting identifies reconstruction-equivalent boundary states. The two values may coincide for a particular realization, but equality requires a cardinality-minimality proof. The role of this realization is therefore not to rename vertex separation. Its role is 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

$$N_{\mathrm{raw}} = 3^4 = 81. \tag{6.15}$$

Let

$$\Phi(C_4) = \{c : V \to \{1,2,3\} : c(u) \neq c(v) \text{ for every } \{u,v\} \in E\} \tag{6.16}$$

be the set of proper three-colorings. The number of admissible colorings is

$$N_{\mathrm{adm}} = |\Phi(C_4)| = 18. \tag{6.17}$$

Fix the partial coloring

$$f(v_1) = 1, \qquad f(v_2) = 2. \tag{6.18}$$

The admissible completions are

$$\big(c(v_3), c(v_4)\big) \in \{(1,2), (1,3), (3,2)\}. \tag{6.19}$$

Hence,

$$N_{\mathrm{rec}}(f) = 3. \tag{6.20}$$

The three cardinalities are

$$N_{\mathrm{raw}} = 81, \qquad N_{\mathrm{adm}} = 18, \qquad N_{\mathrm{rec}}(f) = 3, \tag{6.21}$$

and are pairwise distinct. Here N_{raw} is the cardinality of the complete raw assignment space, N_{adm} is the cardinality of the admissible proper-coloring space, and $N_{\mathrm{rec}}(f)$ is the cardinality of the reconstruction fiber induced by the fixed partial coloring f. These finite cardinalities are not CPR-Width values. In particular, they must not be identified with the active-interface width, the minimum task-sufficient interface size, or the semantic-state count. They are also not direct runtime comparisons. The admissible-to-fiber factor is

$$K_{\mathrm{adm/fiber}} = \frac{18}{3} = 6, \tag{6.22}$$

and the raw-to-fiber factor is

$$K_{\mathrm{raw/fiber}} = \frac{81}{3} = 27. \tag{6.23}$$

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

$$\widetilde{B}_i^{\pi,\tau} = \text{structural candidate interface}$$

from

$$B_i^{\pi,\tau} = \text{cardinality-minimal task-sufficient interface.}$$

The following table summarizes the representative realizations introduced above and developed in the subsequent sections. In the table, the dependence on the

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

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

declared task and reconstruction model is suppressed only for readability.

> A computational problem class can contain a partially reconstructible subclass whenever its instances admit an admissible reconstruction ordering with controlled CPR-Width and satisfy the complete CPR boundary and runtime conditions.

6.5 Common Reconstruction Chain

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

$$\text{Ordering} \longrightarrow \text{Structural Candidate Interface} \longrightarrow \text{Task-Sufficient Interface} \tag{6.24}$$

followed by

$$\text{Reachable Interface States} \longrightarrow \text{Semantic Quotient} \longrightarrow \text{Reconstruction.} \tag{6.25}$$

In symbolic form,

$$\pi \longrightarrow \widetilde{B}_i^{\pi,\tau} \longrightarrow B_i^{\pi,\tau} \longrightarrow \mathcal{U}_i^{\text{reach}}(P, \pi) \longrightarrow \mathcal{S}_i^{\tau}(P, \pi; \mathfrak{M}) \longrightarrow \text{Output}_\tau(P). \tag{6.26}$$

The candidate interface is problem-class dependent. Task sufficiency, minimality, semantic quotienting, and complete runtime accounting are governed by the general CPR-WIDTH theory.

6.6 Tensor Representations

Let

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

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 by

$$\widetilde{B}_i^{T_P,\pi} = \{\alpha \in V_i^\pi : \alpha \text{ still influences an unresolved tensor interaction}\}\,. \quad (6.27)$$

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, it may preserve

- 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 reconstruction-safe signature, then

$$b_i^{\pi,\tau}(T_P) \leq \left|\widetilde{B}_i^{T_P,\pi}\right| \quad (6.28)$$

and

$$\widehat{\omega}_{\mathrm{rec}}(T_P, \pi, \tau; \mathfrak{M}) \leq \max_i \left|\widetilde{B}_i^{T_P,\pi}\right|. \quad (6.29)$$

Tensor representation supplies a structural carrier. It does not by itself prove a small CPR-Width, efficient semantic quotienting, or polynomial runtime.

6.7 Full Adder

The full adder provides a finite output-preserving quotient. Let

$$\mathbb{B} = \{0, 1\}$$

and define the input space

$$X_{\mathrm{FA}} = \mathbb{B}^3.$$

An input state has the form

$$x = (a, b, c_{\mathrm{in}}).$$

The full-adder output map is

$$F_{\mathrm{FA}} : \mathbb{B}^3 \longrightarrow \mathbb{B}^2.$$

Define the Hamming-weight projection

$$C_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) = a + b + c_{\mathrm{in}}. \quad (6.30)$$

Thus,

$$C_{\mathrm{FA}} : \mathbb{B}^3 \longrightarrow \{0, 1, 2, 3\}.$$

Define

$$Q_{\mathrm{FA}}(h) = \left(h \bmod 2, \left\lfloor \frac{h}{2} \right\rfloor\right). \tag{6.31}$$

Then

$$F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}. \tag{6.32}$$

The Hamming-weight fibers have cardinalities

$$1,\ 3,\ 3,\ 1 \tag{6.33}$$

for weights $0, 1, 2, 3$, respectively. Accordingly, the eight input states collapse into four output-relevant equivalence classes:

$$8 \longrightarrow 4. \tag{6.34}$$

This is an exact finite output quotient. The quotient demonstrates semantic compression, but it becomes a CPR realization only after an ordering, an active-interface definition, and a reconstruction transition system have been explicitly specified. It must 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

$$S_{\mathrm{out}}(P_{\mathrm{FA}}) = 4. \tag{6.35}$$

The following chapter may identify this value with a CPR semantic-state count only after specifying the complete reconstruction-stage structure, task, ordering, and model.

6.8 Synthesis of Problem-Class Realizations

The problem classes considered in this chapter possess different structural boundaries. For SAT, the candidate boundary is induced by residual clauses. For CSP, it is induced by unresolved constraint scopes. For graph coloring, it is induced by edges crossing the processed and unresolved vertex sets. For tensor representations, it is induced by unresolved structural interactions. For digital arithmetic, it is induced 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. They become exact minimum CPR interfaces only after cardinality minimality has also been proved. Accordingly, every problem-class realization should establish, in this order,

1. an explicit component representation;
2. a declared task;
3. an admissible reconstruction ordering;
4. a structural candidate interface;
5. a reconstruction-safe signature satisfying S1–S3;
6. task sufficiency of the candidate interface;
7. cardinality minimality, if exact width equality is claimed;
8. semantic-state and transition bounds;

9. and all six complete runtime-cost bounds.

The following chapter develops the staged reconstruction model for digital arithmetic and derives the corresponding CPR-Width and runtime conditions.

7 Finite Verification

Chapter 6 introduced structural realizations of Controlled Partial Reconstruction Width across several problem classes, including the isolated full adder. This chapter records, for a small number of these realizations, exactly what has been verified, at what scope, and what has not been established. Its role is not to introduce new structural theory, but to make explicit the evidentiary status of the constructions already given.

7.1 Purpose of Finite Verification

A mathematical proof of a theorem establishes a claim uniformly, for every member of a stated family, under stated assumptions. A finite verification is different in kind: it is an exhaustive check of a claim on one explicitly stated finite example, or on finitely many such examples. A finite verification does not by itself establish an asymptotic theorem, a complete operation-count advantage, or a measured runtime advantage for a family, and it does not extend automatically to other instances of the same problem class. Both kinds of evidence are used in this manuscript, and they are not interchangeable. Section 7.2 records a finite verification of the isolated full adder: an exhaustive check on its complete eight-state input space. Section 7.3 records a different and stronger kind of result: a proof, valid for every bit-length n, that the ripple-adder output is uniquely and deterministically determined. The distinction between these two kinds of guarantee is maintained throughout this chapter and is made explicit again in Section 7.6.

7.2 Full Adder Verification

Section 6.7 established the Hamming-weight factorization

$$F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ C_{\mathrm{FA}}$$

and the exact output-quotient count

$$S_{\mathrm{out}}(P_{\mathrm{FA}}) = 4.$$

The following table confirms this factorization by exhaustive enumeration of the complete finite input space $\mathbb{B}^3$, rather than by argument alone. For every row,

$$F_{\mathrm{FA}}(x) = Q_{\mathrm{FA}}\Big(C_{\mathrm{FA}}(x)\Big)$$

holds. The exhaustive enumeration verifies, for the complete eight-state input space,

- that the Hamming-weight projection produces exactly four reduced states, so that

$$8 \longrightarrow 4;$$

Table 7.1: Complete finite verification table for the full-adder Hamming-weight factorization.

a	b	c_{in}	h	s	c_{out}	$Q_{\text{FA}}(h)$
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	$(1,1)$

- that the four fibers have cardinalities
$$(1,3,3,1)$$
and partition $\mathbb{B}^3$;
- that Q_{FA} is a bijection onto the image, so the output quotient has exactly four classes;
- and that two input states share an output exactly when they share a Hamming weight.

This record constitutes finite verification evidence: it is checked exhaustively on the complete eight-state input space, but it is not by itself an asymptotic theorem, an operation-count advantage, or a measured runtime advantage.

7.3 Ripple-Adder Verification

The n-bit ripple adder computes
$$(s_0, \ldots, s_{n-1}, c_n)$$
from an encoded input
$$(A, B, c_0)$$
through a sequence of one-bit carry states
$$c_0, c_1, \ldots, c_n \in \{0, 1\},$$
where
$$h_i = a_i + b_i + c_i$$
determines
$$s_i = h_i \bmod 2$$
and
$$c_{i+1} = \left\lfloor \frac{h_i}{2} \right\rfloor$$
at every position i.

Theorem 7.1 (Ripple-Adder Evaluation)**.** *For every $n \geq 1$ 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, and c_0 determine h_0, and hence determine s_0 and c_1 uniquely. Assume inductively that c_i has been computed correctly. Then

$$h_i = a_i + b_i + c_i$$

determines s_i and c_{i+1} uniquely. By induction, every sum bit and the final carry are determined uniquely, for every n. □

This is a proof, not a finite instance check: it holds for every n, not only for one enumerated example.

Proposition 7.1 (No Carry-State Ambiguity)**.** *The one-bit carry signature satisfies the reconstruction-safety conditions S1–S3 introduced in Chapter 2. In particular, two family-reachable prefix states at the same stage with equal carry value produce, for every common next input label* (a_j, b_j)*, the same sum bit and the same successor carry.*

Proof. Let two family-reachable prefix states at stage j have the same carry value c_j. For every common next input label (a_j, b_j), both states therefore produce the same value

$$h_j = a_j + b_j + c_j.$$

Consequently, both states produce the same output bit

$$s_j = h_j \bmod 2$$

and the same successor carry

$$c_{j+1} = \left\lfloor \frac{h_j}{2} \right\rfloor.$$

Thus, equality of the carry signature is preserved under every common next transition and determines the same emitted output information. Hence the carry signature is reconstruction-safe. □

Consequently, the carry state alone is sufficient to continue the reconstruction. No two distinct carry values are conflated, and no additional information from earlier stages is required. There is therefore no state ambiguity in the ripple-adder reconstruction at any stage.

7.4 Projection Fibers

Equations (6.33) and (6.35) already give the fiber cardinalities

$$(1, 3, 3, 1)$$

and the output-quotient count

$$S_{\text{out}} = 4.$$

For completeness of the finite verification record, the four fibers themselves are

$$\Omega_{\text{FA}}(0) = \{(0,0,0)\},$$

$$\Omega_{\text{FA}}(1) = \{(1,0,0),(0,1,0),(0,0,1)\},$$

$$\Omega_{\text{FA}}(2) = \{(1,1,0),(1,0,1),(0,1,1)\},$$

and

$$\Omega_{\text{FA}}(3) = \{(1,1,1)\}.$$

These four fibers are exactly the output-equivalence classes of the isolated full adder: two input states share an output if and only if they lie in the same fiber. The associated binary output-state dimension is

$$D_{\text{out}}(P_{\text{FA}}) = \lceil \log_2 S_{\text{out}}(P_{\text{FA}}) \rceil = 2.$$

This completes the class formation announced in Section 6.7: the eight input states collapse into exactly four classes, and each class is listed explicitly above.

7.5 Verification Protocol

The finite verification records of this chapter distinguish instance-level checks from family-level statements.

Verified models. The isolated full adder is verified at the instance level: its complete eight-state input space is checked exhaustively in Section 7.2. The n-bit ripple adder is verified at the family level: Theorem 7.1 holds for every n, by induction, not by enumeration of finitely many cases.

Verified properties. The verified properties are

- exact output reconstruction from the declared component representation;
- deterministic evaluation at every reconstruction stage;
- and absence of state ambiguity in the carry signature.

Exact results. All results in this chapter are exact within their stated scope: either an exhaustive statement about eight input states, or a proof valid for all n under the stated staged model. No approximation or heuristic is involved, and no probabilistic argument is used.

Limits of verification. These results do not extend to other problem classes by analogy. In particular,

$$L^* \in \mathbf{P}_{\text{CPR}} \Longrightarrow \mathbf{P} = \mathbf{NP}$$

for an NP-complete language L^* remains strictly conditional: no NP-complete language is shown in this manuscript to belong to $\mathbf{P}_{\text{CPR}}$. The finite and structural results recorded here do not by themselves establish tractability for unrestricted problem classes such as general SAT, CSP, or tensor instances.

7.6 Evidence Level

Results in this manuscript are classified by the type of justification actually provided, not by a single linear hierarchy of confidence.

Finite verification evidence.
A claim is checked exhaustively on one explicitly stated finite example.
Structural evidence.
A recurring structural pattern is identified across several models or examples, without a full asymptotic proof.
Asymptotic theorem evidence.
A theorem is proved uniformly for an entire family, under explicit assumptions.

On this basis, the realizations of Chapter 6 fall into three groups.

Mathematically fully verified.
The full adder (Section 7.2: exhaustive finite verification) and the ripple adder (Section 7.3: uniform proof for all n).
Exemplarily verified.
Graph coloring on C_4 (Section 6.3, Quantitative Verification for C_4): one concrete instance is worked out in full, but no family-level certification is given.
Theoretically formulated.
Boolean satisfiability, constraint satisfaction, and tensor representations (Chapter 6): structural candidate interfaces and reconstruction templates are given, but no finite verification record or asymptotic theorem is established for these classes in this manuscript.

A finite verification does not establish an asymptotic theorem, and an asymptotic theorem for one family does not extend to another problem class. Every result in this manuscript should be read together with the specific evidence level stated for it.

Appendix A

Core Symbol Register

This appendix records the main symbols used throughout the manuscript. The purpose is to provide 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.
$\omega_{\text{rec}}(P, \pi)$	Controlled Partial Reconstruction Width induced by π.
$\omega_{\text{rec}}(P)$	Minimum Controlled Partial Reconstruction Width over admissible orderings.
$\Pi_{\text{adm}}(P)$	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.
$\mathcal{U}_i^{\text{reach}}(P, \pi)$	Reachable interface-state space at stage i.
$\mathcal{E}_i(u)$	Admissible future-continuation set of a reachable state u; task-independent.
$\Theta_{P,i}^\tau$	Task-evaluation map; extracts the task-relevant information from a continuation set.
$\nu_{P,i}^\tau$	Signature-normalization map; assigns a canonical representative to a task-evaluation value.
$\widehat{\sigma}_{P,i}^\tau(u)$	Normalized future signature of u: $\widehat{\sigma}_{P,i}^\tau(u) = \nu_{P,i}^\tau\Big(\Theta_{P,i}^\tau(\mathcal{E}_i(u))\Big)$.
$\equiv_{P,i}^\tau$	Task-relative signature equivalence: $u \equiv_{P,i}^\tau v \iff \widehat{\sigma}_{P,i}^\tau(u) = \widehat{\sigma}_{P,i}^\tau(v)$.
$\mathcal{S}_i(P, \pi, \tau)$	Semantic quotient state space at stage i.
$S_{\text{CPR}}(P, \pi, \tau)$	Maximum semantic-state count of the reconstruction process.
N_{CPR}	Total CPR operation count.
N_{Build}	Operations required to build the initial representation.

Continued on the next page

Symbol	Meaning
N_{Reduce}	Operations required for reduction and semantic quotienting.
$N_{\text{Transition}}$	Operations required for stage-to-stage transitions.
$N_{\text{Reconstruct}}$	Operations required to reconstruct admissible outputs.
N_{Verify}	Operations required to verify reconstructed outputs.
N_{Output}	Operations required to produce final outputs.
G_{count}	Elementary-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 $\mathcal{F}$.
P_{CPR}	Class of languages certified by the CPR-WIDTH audit conditions.
P	Classical polynomial-time decision class.
NP	Classical nondeterministic polynomial-time decision class.

Bibliography

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge: Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.

[2] Hans L. Bodlaender. "A Partial k-Arboretum of Graphs with Bounded Treewidth". In: *Theoretical Computer Science* 209.1–2 (1998), pp. 1–45. DOI: 10.1016/S0304-3975(97)00228-4.

[3] Stephen A. Cook. "The Complexity of Theorem-Proving Procedures". In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. STOC '71. New York, NY, USA: Association for Computing Machinery, 1971, pp. 151–158. DOI: 10.1145/800157.805047.

[4] Marek Cygan et al. *Parameterized Algorithms*. Cham: Springer, 2015. ISBN: 978-3-319-21274-6. DOI: 10.1007/978-3-319-21275-3.

[5] Adnan Darwiche and Pierre Marquis. "A Knowledge Compilation Map". In: *Journal of Artificial Intelligence Research* 17 (2002), pp. 229–264. DOI: 10.1613/jair.989.

[6] Rina Dechter. *Constraint Processing*. San Francisco: Morgan Kaufmann, 2003. ISBN: 978-1-55860-890-0.

[7] Reinhard Diestel. *Graph Theory*. 5th ed. Vol. 173. Graduate Texts in Mathematics. Berlin: Springer, 2017. ISBN: 978-3-662-53621-6. DOI: 10.1007/978-3-662-53622-3.

[8] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. London: Springer, 2013. ISBN: 978-1-4471-5558-4. DOI: 10.1007/978-1-4471-5559-1.

[9] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman, 1979. ISBN: 978-0-7167-1045-5.

[10] David Money Harris and Sarah L. Harris. *Digital Design and Computer Architecture*. 2nd ed. Waltham, Massachusetts: Morgan Kaufmann, 2012. ISBN: 978-0-12-394424-5.

[11] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations*. Ed. by Raymond E. Miller and James W. Thatcher. New York: Plenum Press, 1972, pp. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

[12] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2009), pp. 455–500. DOI: 10.1137/07070111X.

[13] Christos H. Papadimitriou. *Computational Complexity*. Reading, Massachusetts: Addison-Wesley, 1994. ISBN: 978-0-201-53082-7.

[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. DOI: 10.1016/0196-6774(86)90023-4.

[15] Francesca Rossi, Peter van Beek, and Toby Walsh, eds. *Handbook of Constraint Programming*. Vol. 2. Foundations of Artificial Intelligence. Amsterdam: Elsevier, 2006. ISBN: 978-0-444-52726-4.

[16] Michael Sipser. *Introduction to the Theory of Computation*. 3rd ed. Boston: Cengage Learning, 2012. ISBN: 978-1-133-18779-0.