Exact Structural Abstraction and Tractability Limits

Tristan Simas
McGill University
tristan.simas@mail.mcgill.ca

April 14, 2026

Abstract

Any rigorously specified problem determines an admissible-output relation R, and the only state distinctions that matter are the classes $s \sim_R s' \iff \mathrm{Adm}_R(s) = \mathrm{Adm}_R(s')$. Every exact correctness claim reduces to the same quotient-recovery problem, and the no-go concerns tractability of the underlying problem, not of its presentation. Exact means agreement with R, not zero-error determinism or absence of approximation/randomization in the specification. The exact-semantics quotient theorem identifies admissible-output equivalence as the canonical object recovered by exact relevance certification. Decision, search, approximation, statistical, randomized, horizon, and distributional guarantees instantiate it. Tractable families have a finite primitive basis, but optimizer-quotient realizability is maximal, so quotient shape cannot characterize the frontier.

We prove a meta-impossibility theorem for efficiently checkable structural predicates invariant under theorem-forced closure laws of exact certification. Zero-distortion summaries, quotient entropy bounds, and support counting explain them. Same-orbit disagreements across four obstruction families, via action-independent pair-targeted affine witnesses, force contradiction. Consequently no correct problem-tractability classifier on a closure-closed domain yields an exact characterization over these families. Restricting to a closure-closed subdomain helps only by removing orbit gaps. Uniform strict-gap control preserves the full optimizer quotient, while arbitrarily small perturbations can flip relevance and sufficiency. Closure-orbit agreement is forced by correctness, and the same compute-cost barrier extends to optimizer computation, payload/search, and theorem-backed external or transported outputs. The obstruction therefore appears at the level of correctness itself, not any particular output formalism.

1 Introduction

Any rigorously specified computational problem already determines a state-indexed admissible-output relation R, and the only state distinctions that matter are the admissible-output equivalence classes $s \sim_R s' \iff \mathrm{Adm}_R(s) = \mathrm{Adm}_R(s')$. Informally, knowing what matters is the only thing that matters. Every exact correctness claim reduces to the same quotient-recovery problem. Here exact means exact agreement with the admissible-output relation itself, not zero-error determinism or the absence of approximation, randomization, statistical thresholds, or failure states inside the specification. Section 6 proves this directly: Boolean payloads and exact predicates transfer definitionally to exact relevance certification, sufficiency is relation refinement of the admissible-output quotient, relevance is erased failure of that refinement, exact relevance certification for the induced decision problem realizes exactly the same quotient structure, every state equivalence relation is realizable at this semantic level, and every correctness claim admits a coordinate presentation (Corollaries 6.26 and 6.27, Proposition 6.32, Corollary 6.33, Proposition 6.31, Theorem 6.34, Proposition 6.36, and

Corollary 6.50). Decision, counting, search, approximation, PAC/regret/risk, randomized-output, finite-horizon or anytime, and distributional guarantees therefore all reduce to the same quotient-recovery problem. The optimizer-quotient framework of [25] isolates this canonical semantic layer, and the earlier optimizer-quotient factorization results are recovered as verification of the same spine.

The validity relation is the correctness condition itself, not an optional encoding trick. Different coordinate presentations may make quotient recovery trivial or rich, but they do not change the canonical quotient itself. Proposition 6.36 gives the universal one-coordinate realization, while Propositions 6.37–6.39 give structurally informative realizations of the same semantics. SAT fits through its exact yes/no validity predicate, sorting through the set of correctly sorted outputs for each input, and fixed approximation or PAC guarantees through the outputs meeting the stated threshold. The frontier theorems therefore concern tractable recovery of a canonical quotient across presentations, not existence of semantics.

The orbit-gap contradiction depends only on closure-law invariance (Theorem 4.9), and closure-law invariance is itself forced by correctness (Theorem 6.17). Closure-equivalent representations encode the same certification problem, so a correct tractability classifier must assign them the same verdict. The verdict concerns tractability of the underlying problem, not tractability of a specification format. The same forced-agreement statement extends to polynomial-time solvability of optimizer computation, canonical payload and search tasks, external output objects, and representation-relative transported outputs such as policies and randomized procedures, so the no-go applies uniformly to the compute-cost layer as well as to feature sufficiency. Restricting the domain is therefore not a loophole in kind: Corollary 4.14 shows that it helps only if the restricted closure-closed domain eliminates all orbit gaps for the target notion. The specific admissibility package of Definition 6.16 is one natural instantiation, but the package itself is not load-bearing.

This is a structural invariance barrier: once correctness forces closure-orbit agreement, no finite admissible classifier can separate the obstruction families.

Definition 6.16 is not a technical convenience for manufacturing a no-go theorem. Closure-invariance is forced by correctness (Theorem 6.17), and the remaining clauses isolate the direct structural-classification regime: efficiently checkable, presentation-independent, structurally extractable verdicts rather than re-encodings of the original semantic problem under structural names. A procedure that violates these guardrails has not produced a direct structural frontier theorem; it has relocated the difficulty into hidden global computation or richer invariants.

On the positive side, we first show that all known tractable cases collapse to eight primitive mechanisms, together with regime lifts and degenerate collapses. This gives a finite inventory of the positive side, but not a frontier theorem: optimizer realizability is maximal, so quotient shape alone is too expressive to classify tractability.

Exactness is not arbitrary. At zero distortion, any summary that preserves optimal actions must still separate distinct optimizer classes: if two different optimizer classes were assigned the same summary value, the summary would merge states with different optimal-action sets and would cease to be lossless. Exact relevance therefore marks the theoretical floor for lossless abstraction. Section 6 now also fixes the approximation boundary: approximate surrogate claims about relevance or sufficiency are admissible only with an explicit stability reduction to the exact optimizer sets, because witness-gap control preserves the corresponding witness, a uniform strict-gap hypothesis preserves the entire optimizer quotient together with the sufficient, relevant, and minimal-sufficient structure, and arbitrarily small uniform perturbations can otherwise flip the exact judgment.

The realizability barrier forces the organizing principle: tractability predicates must be evaluated on representation-level structural classes, but modulo theorem-forced equivalences of presentation. In particular, closure under relabelings, positive affine utility reparameterizations, duplication, and

binary irrelevant-coordinate extension is not a modeling preference; it is inherited from the fact that exact certification depends only on the induced decision quotient relation.

Standing Semantics

For a decision problem with state space S, action space A, and utility U, write

$$\mathrm{Opt}(s) = \{a \in A : U(a, s) \text{ is maximal among all actions at } s\}.$$

This induces the decision-equivalence relation

$$s \sim_{\mathrm{Opt}} s' \quad \Longleftrightarrow \quad \mathrm{Opt}(s) = \mathrm{Opt}(s').$$

The associated optimizer quotient is the quotient set $S/\!\sim_{\mathrm{Opt}}$. A coordinate set I is *sufficient* when agreement on I forces equality of optimizer classes, and a coordinate is *relevant* when deleting it destroys sufficiency. Later closure laws and obstruction families are all phrased relative to this quotient viewpoint.

At the same time, unrestricted predicates trivialize any impossibility theorem via oracle encoding. We therefore isolate an admissibility layer: polynomial-time checkability, structural extractability, closure-law invariance, and bounded-pattern definability. This class excludes direct semantic encoding while still covering the structural results established here. The proof-load-bearing clause is closure-law invariance (Theorem 4.9); the remaining clauses act as guardrails that keep the search space structural rather than oracle-like.

This is a no-go for classifiers inside the admissibility class, not a no-go for tractable classes themselves. Stronger representation-sensitive structure can lie outside Definition 6.16; for example, the algebraic invariants used in CSP dichotomy theorems are not bounded-pattern definable in the present sense. This is why CSP-style positive frontier theorems are compatible with the present result rather than in tension with it. The open problem is therefore not to evade the theorem, but to identify a successor admissibility class that uses stronger structure of that kind.

Within this admissibility class, we prove a four-family no-go template. For each obstruction family we construct two slices with different obstruction status in the same closure orbit. The key mechanism is an action-independent, pair-targeted additive affine term (the statewise "α-component" of positive affine transport), which changes the obstruction predicate while preserving closure equivalence. Closure-law invariance then transports any candidate predicate across the orbit and forces contradiction.

The dominant-pair family is the canonical instance, and the same mechanism now applies to margin masking, ghost-action concentration, and additive/statewise offset concentration. Theorem 4.9 identifies closure-law invariance as the only load-bearing hypothesis, and Theorem 6.17 upgrades the scope from invariant classifiers to correct classifiers on closure-closed domains. The remaining admissibility clauses serve as guardrails. The open problem is to identify a stronger structural principle, if one exists, that yields a correct frontier theorem.

An accompanying Lean development mechanically verifies the realizability constructions, the generic orbit-gap argument, and the family-level admissible no-go theorems used in the collapse argument; the archived artifact is available at `https://doi.org/10.5281/zenodo.19457896`.

2 Finite Basis for the Current Positive Landscape

We distinguish between *tractable families treated here* and *primitive tractability mechanisms*. A tractable family is any named positive condition included in the present analysis. A primitive

mechanism is a smaller conceptual source of tractability that may explain several families at once. The positive theory organizes into three roles: six core structural families, four regime lifts, and five degenerate collapses. Projecting those families to primitive mechanisms leaves eight sources of tractability: the six core subcases together with constant-optimizer collapse and finite explicit enumeration. These eight mechanisms are the explicit inventory of the current positive landscape. The universal theorem is quotient universality for exact correctness; the later no-go shows that no finite admissible classifier extends this positive inventory to an exact frontier theorem.

Theorem 2.1[1] (Explicit Partition of the Current Positive Landscape). *The fifteen tractable families considered here admit an explicit disjoint partition into three blocks:*

1. *six* core *structural families,*
2. *four* lifted *families that reduce to existing core subcases, and*
3. *five* degenerate *families that become tractable by optimizer collapse or finite explicit enumeration.*

Moreover, each family is classified into exactly one of these three blocks by an explicit classification map.

Proposition 2.2[2] (Explicit Inventory). *The fifteen tractable families considered here are exactly the following:*

1. *core: bounded actions, separable utility, low tensor rank, tree structure, bounded treewidth, coordinate symmetry;*
2. *lifts: product distribution, bounded support, bounded horizon, full observability;*
3. *degenerate: single action, bounded state space, strict global dominance, constant optimal set, multiplicative-separable constant-sign.*

Corollary 2.3[3] (Finite Basis for the Current Positive Landscape). *Projecting the partition of Theorem 2.1 to primitive mechanisms yields an explicit finite list of eight primitive tractability mechanisms. The list consists of the six core structural subcases*

bounded actions, *separable utility*, *low tensor rank*, *tree structure*, *bounded treewidth*, *coordinate symmetry*,

together with two degenerate mechanisms:

constant-optimizer collapse and *finite explicit enumeration.*

Proposition 2.4[4] (Explicit Primitive Basis). *The current primitive basis consists exactly of bounded actions, separable utility, low tensor rank, tree structure, bounded treewidth, coordinate symmetry, constant-optimizer collapse, and finite explicit enumeration.*

The current positive landscape therefore contains fifteen tractable families accounted for by eight primitive mechanisms.

Proposition 2.5[5] (All Independent Core Mechanisms Are Nondegenerate). *The analysis already contains witness families showing that five core mechanisms are individually indispensable in the sense of basis coverage:*

1. *a low-rank-only witness family,*

[1] Lean: `FR1-2` [2] : `FR25-28` [3] : `FR3` [4] : `FR29` [5] : `FR101`, `FR105-106`, `FR108`, `FR112-113`

2. *a separable-only witness family,*

3. *a bounded-actions-only witness family,*

4. *a bounded-treewidth-only witness family, and*

5. *a symmetry-only witness family.*

Each of these families remains tractable by its designated mechanism while failing the corresponding versions of the other independent core mechanisms.

Proposition 2.5 settles the independent part of the basis question on the present surface. Every core mechanism that is not merely contained in another is shown to be genuinely needed.

Concretely, the present theory contains five kinds of "only-by" witness: low-rank only, separable only, bounded-actions only, bounded-treewidth only, and symmetry only.

Proposition 2.6[6] (Weak Tree Ordering Does Not Imply Width One). *There exists a dependency presentation satisfying the older monotone tree-order predicate* `TreeStructured` *whose dependency graph has real treewidth greater than one. Width-one containment requires the stricter parent-tree condition from Proposition 2.8, not merely monotone ordering.*

Proposition 2.7[7] (Cyclic Dependencies Recover Hardness). *The reduction family for exact relevance certification yields coNP-hard instances with cyclic dependency structure. Thus the tree-structured regime is not only sufficient for tractability; leaving it also reaches the hard side.*

Proposition 2.8[8] (Parent-Tree Structure Gives Width-One Decompositions). *Suppose a dependency presentation is tree-structured in the strict parent sense: every dependency points to a smaller coordinate and every coordinate has at most one parent. Then the induced dependency graph admits a genuine tree decomposition of width at most one.*

Proposition 2.8 is the containment theorem behind the intended "tree structure is the treewidth-one special case" slogan. It distinguishes this parent-tree notion from the weaker monotone-order predicate currently named `TreeStructured`.

Proposition 2.9[9] (Current Positive Interfaces Factor Through Role Classification). *The current positive theorem surface factors through the role classifier of Theorem 2.1. Concretely, the family-indexed interfaces for*

1. *subcase-facing tractability,*

2. *degenerate-case witnesses, and*

3. *complexity-class summaries*

all factor through the map

$$\mathit{family} \longmapsto \mathit{role} \in \{\mathit{core}, \mathit{lifted}, \mathit{degenerate}\}.$$

Proposition 2.9 shows that the exported positive API is already controlled by the role partition rather than by fifteen unrelated theorem schemas.

3 Lifts and Degeneracies

The extra named families beyond the six core subcases split naturally into two groups.

[6] : `FR114` [7] : `FR115` [8] : `FR100`, `FR110-111` [9] : `FR4`

Regime Lifts

Four stochastic/sequential positive results do not introduce new primitive mechanisms. They reduce directly to existing core subcases:

1. product distributions reduce to separable utility;
2. bounded support reduces to bounded actions;
3. bounded horizons reduce to bounded treewidth;
4. full observability reduces to tree structure.

These are mathematically useful because they transfer the static tractable theory into richer regimes. But they do not enlarge the primitive basis. They are best understood as *lifts*, not new frontier points.

Proposition 3.1[10] (Lifted Cases Introduce No New Primitive). *Each stochastic/sequential tractable case considered here is classified into one of the six core structural subcases. In particular, these positive results do not enlarge the primitive basis from Corollary 2.3.*

Degenerate Cases

The remaining positive cases considered here are degenerate in a precise sense.

Single-action systems, strict global dominance, constant optimal-set systems, and multiplicative-separable constant-sign models all collapse the optimizer so strongly that the relevant-coordinate question disappears: the optimal-action set is constant across states, so the empty coordinate set is sufficient. These are not new structural tractability mechanisms for the relevance problem; they are constant-optimizer collapses.

Bounded state space behaves differently. It does not force a constant optimizer. Instead, it makes exact certification feasible by brute-force enumeration over a finite universe. This is polynomial only because the instance family already carries a hard external bound on its search space.

Proposition 3.2[11] (Degenerate Positive Cases). *Every non-core tractable case considered here that is not a regime lift belongs to one of two degenerate buckets:*

1. *constant-optimizer collapse, or*
2. *finite explicit enumeration.*

Taken together with Proposition 3.1, Proposition 3.2 sharpens Theorem 2.1: every positive result considered here is exactly one of three things, a primitive core case, a lift, or a degenerate collapse.

4 Toward the Frontier Theorem

Sections 2 and 3 motivate a frontier question. Do the tractable families already suggest a finite optimizer-compatible criterion? In the strongest form, could exact relevance certification admit a representation-sensitive dichotomy theorem analogous to the classical dichotomy programs for satisfiability and constraint satisfaction?

The most direct version of that program fails. Quotient shape is too expressive, and later in the section even structurally restricted finite classifiers fail on a theorem-forced surface. Any future frontier theorem must therefore avoid both sources of failure.

[10] : `FR5` [11] : `FR1`, `FR6`

Three obstacles remain. Realizability is not the bottleneck: Theorem 5.1 shows that arbitrary labeling kernels already arise as optimizer quotients, so no frontier theorem can be driven by quotient realizability alone. Even on the positive side one must separate primitive mechanisms from lifts and degeneracies; otherwise a finite inventory confuses genuine sources of tractability with derived or collapsed cases. And although the long-term target remains an algebraic description analogous to the CSP dichotomy program, the negative results below show that no direct optimizer-compatible classifier can play that role.

A Stabilized Binary Pairwise Subregime

One piece of the frontier program already admits a clean canonical theorem. On binary coordinate domains, define the mixed difference along coordinates i, j for action a by evaluating the utility on the four assignments $(0,0),(1,1),(0,1),(1,0)$ while holding all other coordinates fixed at 0. For pairwise utilities, this mixed difference is the canonical witness of genuine pair interaction.

Proposition 4.1[12] (Binary Pairwise Symmetry Dichotomy). *Let the coordinate alphabet be binary. If a utility admits a pairwise decomposition and is invariant under coordinate permutations, then either:*

1. *it admits a unary-coordinate decomposition, so every pairwise term collapses into single-coordinate contributions, or*
2. *every distinct coordinate pair carries a genuine pair interaction, witnessed by a nonzero mixed difference for some action.*

Within the binary pairwise-symmetric regime, there is no intermediate sparse interaction pattern hiding between unary collapse and the complete-pair case. Once a single nonzero pair witness exists, symmetry propagates it to every coordinate pair. This removes one plausible source of additional primitive mechanisms from the frontier search.

The optimizer-relevant version needs one extra layer. Raw pair interaction is still too coarse, because action-independent pairwise terms can be structurally dense without changing the optimizer. The correct invariant is the mixed difference of action gaps $U(a,\cdot) - U(b,\cdot)$.

Proposition 4.2[13] (Decision-Relevant Binary Pairwise Dichotomy). *In the same binary pairwise-symmetric regime, either all action-dependent pair effects collapse into a unary-coordinate reduction after removing an action-independent base state term, or every distinct coordinate pair is decision-relevantly interacting for some action gap.*

A coarse hardness heuristic therefore fails. The obstruction theorem shows that even complete genuine pair interaction and failure of coordinate symmetry do not force hardness by themselves: one can still have a constant optimizer. Any eventual frontier theorem must therefore quotient out optimizer-degenerate phenomena, not just declared or utility-level interaction density.[14]

The obstruction is stronger than that first formulation suggests. Action-specific utility offsets leave the decision-relevant interaction graph unchanged, yet they can turn a family with nonconstant optimizer behavior into one with a constant optimizer while preserving dense decision-relevant structure. Any true hardness theorem in this regime must therefore normalize away action offsets explicitly, not merely require dense decision-relevant interaction.[15]

Identify utilities up to action-dependent, state-independent offsets, and the offset-normalized decision-relevant graph becomes well-defined on those equivalence classes. But the resulting dichotomy attempt still fails. The offset-normalized obstruction theorem shows that even after pass-

[12] : `FR118-120` [13] : `FR121-124` [14] : `FR124` [15] : `FR125-127`

ing to offset classes, one can retain complete decision-relevant interaction and unbounded treewidth while exact certification remains trivial.[16]

The next refinement is to restrict the interaction graph to optimizer-supported actions. This removes two earlier collapse mechanisms: the offset-collapse family and a ghost-action family whose all-action decision-relevant graph is complete even though a single coordinate remains sufficient. But even that support-filtered graph is not enough. The optimizer-supported obstruction theorem exhibits a margin-masking family with complete optimizer-supported decision-relevant interaction and unbounded treewidth, yet the optimizer still depends only on coordinate 0. The third collapse mechanism is therefore not ghost support or offset collapse, but large unary margins that mask dense supported pairwise interactions.[17]

One might hope that a strict unary-to-pair margin bound rescues the dichotomy. Define *margin-bounded* pairwise utilities by requiring every unary term to be at most twice the largest binary mixed-difference magnitude. This still fails. The dominant-pair family is margin-bounded because its unary terms vanish, while its optimizer-supported decision-relevant graph remains complete. Yet the obstruction theorem shows that exact certification is still controlled by the two-coordinate set $\{0, 1\}$. The fourth collapse mechanism is therefore pair-weight concentration rather than unary masking: one supported pair dominates the optimizer while the remaining dense interactions are too weak to matter.[18]

Admissible Closure-Orbit No-Go Program

The obstruction families above lead to a uniform negative statement because they admit explicit disagreements inside a single closure orbit.

Proposition 4.3[19] (Orbit-Gap Template). *Let Q be a slice predicate. Suppose there exist slices U, V such that U and V lie in the same closure orbit, $Q(U)$ holds, and $Q(V)$ fails. Then no closure-law-invariant predicate P can satisfy*

$$P(W) \iff Q(W) \qquad \text{for all slices } W.$$

Proof idea. Closure-law invariance gives $P(U) \iff P(V)$. Exact agreement with Q would then imply $Q(U) \iff Q(V)$, contradicting the assumed orbit gap. ■

For each obstruction family, the argument constructs two same-orbit slices with different obstruction status. In all four cases the witness is a positive-affine step whose α-component is an action-independent state term supported on a single coordinate pair. Pure relabeling cannot change the relevant obstruction statistics, and a global scale factor preserves all pairwise magnitudes up to common rescaling. The statewise additive term is what allows one to move mass onto a chosen pair while remaining inside the theorem-forced closure laws. The same mechanism works for four structurally different obstruction families, and Proposition 4.12 together with Corollary 4.13 shows that orbit gaps are the complete obstruction criterion in the closure-invariant and correctness-forced regimes.

There is also a transport interpretation of these witnesses. Once transport is computed on optimizer-quotient classes, a singleton quotient admits zero-cost diagonal transport, whereas genuine branching forces positive off-diagonal transport when distinct classes carry mass. The pair-targeted affine witnesses exploit exactly this sensitivity. They do not introduce arbitrary perturbations unrelated to the decision geometry; rather, they move mass across a selected branch of the quotient while staying inside the same closure orbit. The affine witnesses therefore track the intrinsic transport geometry of the quotient instead of functioning as purely syntactic tricks.[20]

[16] : FR128-129 [17] : FR130-136 [18] : FR137-140 [19] : FR185 [20] : FR183-184

Theorem 4.4[21] (Dominant-Pair Admissible No-Go). *No admissible normalization predicate can decide the dominant-pair obstruction family exactly.*

Proof sketch. The target predicate is unique dominant-pair status: one pair/action realizes the maximal interaction magnitude and that maximizer is unique. Start from a slice whose unique dominant pair is $\{0, 1\}$. Add an action-independent affine state term supported on a different pair, say $\{1, 2\}$. This produces a slice in the same closure orbit but with a different unique dominant pair. Proposition 4.3 then rules out every closure-law-invariant classifier, hence every admissible one. ■

Theorem 4.4 gives the first instance. The remaining three families use the same orbit-gap scheme, but each isolates a different structural obstruction.

At the compute-cost layer, the same dominant-pair orbit witness yields a machine-checked no-go for optimizer computation: no correct classifier for polynomial-time solvability of optimizer computation can decide dominant-pair status exactly. In particular, no admissible predicate can simultaneously track optimizer-computation polynomiality and exact dominant-pair status.[22]

Theorem 4.5[23] (Margin-Masking Admissible No-Go). *No admissible normalization predicate can decide margin-boundedness exactly.*

Proof sketch. For margin masking, the target predicate is margin-boundedness itself. This family is structurally different from the dominant-pair family because it is governed by a threshold comparing unary mass against the largest pair interaction. The affine witness leaves unary terms unchanged but raises the largest pair interaction past the relevant threshold, so the translated slice becomes margin-bounded while the base slice is not. Proposition 4.3 therefore rules out every closure-law-invariant classifier, hence every admissible one. ■

Theorem 4.6[24] (Ghost-Action Admissible No-Go). *No admissible normalization predicate can decide the ghost-action concentration signature exactly.*

Proof sketch. For ghost actions, the target predicate says that there is an action whose unary contribution on the first coordinate is -1 on both binary values and whose anchor-pair interaction has unit mixed-difference magnitude. This family is local in appearance but unstable under the same pair-targeted affine move. A pair-supported affine term preserves the closure orbit while destroying the signature, so Proposition 4.3 applies. ■

Theorem 4.7[25] (Offset Admissible No-Go). *No admissible normalization predicate can decide the additive/statewise offset signature exactly.*

Proof sketch. For additive/statewise offset concentration, the target predicate requires two actions with distinct anchor-pair interaction magnitudes, namely 1 and 0. This family isolates the action-specific mismatch that survives after earlier offset normalizations. A pair-supported affine term changes the anchor-pair statistics while remaining in the same closure orbit. Again, Proposition 4.3 yields the contradiction. ■

Theorem 4.8[26] (Admissible Collapse Across the Four Obstruction Families). *Let P be a normalization predicate satisfying the admissibility axioms. Then P is not a tractability characterization.*

[21] : FR186 [22] : FR250-251 [23] : FR187 [24] : FR188 [25] : FR189 [26] : FR190

Proof idea. If P were an admissible tractability characterization, it would have to decide each of the four obstruction predicates above. Theorem 4.4, Theorem 4.5, Theorem 4.6, and Theorem 4.7 show that this is impossible. Hence no admissible normalization predicate yields the desired frontier theorem. ■

The four obstruction families are witnesses for the orbit-gap template (Proposition 4.3), not the scope of Theorem 4.8. The theorem ranges over the entire admissibility class. Four structurally distinct families are included because the same pair-targeted affine transport defeats classifiers across families that share no surface structure, and Proposition 4.12 shows that orbit gaps are the complete obstruction criterion for exact classification by closure-law-invariant predicates. The four families are therefore theorem-backed witnesses of a complete mechanism rather than an ad hoc list.

Abstractly, let Γ be a class of slice predicates, and call a predicate Γ-admissible when it belongs to Γ. Call Γ *closure-sound* if every Γ-admissible predicate is closure-law invariant. Call Γ a *reasonable guardrail package* if it is closure-sound and also imposes auxiliary restrictions such as efficient checkability, structural extractability, and bounded-pattern definability. The proof below uses only closure-soundness; the remaining clauses explain which non-oracular search spaces the no-go is meant to cover.[27]

Theorem 4.9[28] (Closure-Sound Package No-Go). *Let Γ be any class of slice predicates such that every Γ-admissible predicate is closure-law invariant. Then no Γ-admissible predicate can simultaneously characterize the four obstruction families. In particular, the no-go theorem depends only on closure-law invariance; the remaining clauses of Definition 6.16 justify the admissibility package as a meaningful notion but do not enter the contradiction.*

Proof sketch. Closure-law invariance suffices to invoke Proposition 4.3 against each of the four obstruction families. The same orbit-gap proofs underlying Theorem 4.4, Theorem 4.5, Theorem 4.6, and Theorem 4.7 then yield the contradiction directly. Any one of the four families independently witnesses the collapse; we state all four to make explicit that the obstruction is robust across structurally distinct witnesses. ■

At the Γ-generic level, the formal contradiction is simultaneous: one Γ-admissible predicate cannot characterize all four families at once. The individual-family contradictions above are formalized separately for the specific admissibility package of Definition 6.16.

Theorem 4.9 is the load-bearing form of Theorem 4.8. Any reasonable formalization of admissibility, whether the specific package of Definition 6.16 or any alternative whose admissible predicates are closure-law invariant, inherits the impossibility conclusion. Proposition 6.1 identifies closure-law invariance as the semantic core, and Theorem 6.17 shows that correctness forces the same invariance on any tractability classifier. The admissibility package is one natural instantiation; the theorem is robust to its replacement.

Corollary 4.10[29] (Reasonable Guardrail Packages). *The same conclusion holds for every reasonable guardrail package. In particular, efficient checkability, structural extractability, bounded-pattern definability, and the informal exclusion of oracle encodings are guardrails rather than proof-load-bearing hypotheses.*

Proof. Immediate from Theorem 4.9, since only the closure-soundness clause is used. ■

Corollary 4.11[30] (No Correct Tractability Classifier). *Let C be any correct tractability classifier on a domain that is closed under the closure laws and contains the four obstruction families. Then C cannot yield an exact tractability characterization across those families. In particular, no procedure*

[27] : FR191-192 [28] : FR193 [29] : FR194 [30] : FR193, FR197-199

assigning verdicts to representations can correctly predict tractability across the four obstruction families, regardless of whether the procedure tracks closure-invariant features internally.

Proof. By Theorem 6.17, any such classifier must agree on closure orbits and is therefore closure-law invariant on its domain. Theorem 4.9 then rules out exact characterization across the four obstruction families. ■

Corollary 4.11 is the witness-level universal form of the no-go. Theorem 4.9 isolates closure-law invariance as the load-bearing hypothesis, and Theorem 6.17 removes even that as an explicit assumption by forcing the same invariance from correctness itself. Corollary 4.14 then states the general domain-relative form: restricting the domain helps only if it removes all orbit gaps for the target notion.

Proposition 4.12[31] (Orbit-Gap Completeness for Exact Classification). *For any slice predicate Q, the following are equivalent:*

1. *there exists a closure-law-invariant predicate P such that $P(S) \iff Q(S)$ for every slice S;*
2. *Q is itself closure-law invariant;*
3. *Q is constant on closure orbits.*

Equivalently, exact classification of Q by closure-law-invariant predicates fails if and only if Q admits an orbit-gap witness: two closure-equivalent slices with different Q-status.

Proof sketch. The implication (1) ⇒ (2) transfers closure-law invariance across exact equivalence. The implication (2) ⇒ (3) says exactly that a closure-law-invariant predicate cannot change value inside a closure orbit. For (3) ⇒ (2), every primitive closure-law step is already a closure equivalence, so orbit constancy supplies the six invariance clauses. The final equivalence is the contrapositive of orbit constancy. ■

Orbit gaps are the complete obstruction criterion for exact classification of any fixed target predicate by closure-law-invariant classifiers. The four obstruction families are not claimed to exhaust all hard-side phenomena; they provide structurally distinct witnesses of that criterion.

Corollary 4.13[32] (Orbit-Gap Completeness on Closure-Closed Domains). *Let D be a closure-closed domain of slices, and let T be a target predicate on D. Then T admits an exact characterization on D by a closure-law-invariant predicate if and only if T has no orbit-gap witness inside D. Equivalently, exact characterization on D by closure-law-invariant predicates fails if and only if there exist $U, V \in D$ in the same closure orbit with different T-status.*

Proof sketch. This is the domain-restricted version formalized in Lean. The forward implication is immediate from closure-law invariance. For the reverse implication, take the closure hull of $D \cap T$; if T has no orbit-gap witness inside D, that closure hull agrees with T on D. ■

Applied to exact tractability, Theorem 6.17 places every correct tractability classifier on a closure-closed domain inside this regime. Orbit gaps are therefore the complete obstruction criterion for the universal-scope no-go once correctness is imposed.

[31] : `FR201-204` [32] : `FR210-213`

Corollary 4.14[33] (Domain Restriction Helps Only by Removing Orbit Gaps). *Let D be a closure-closed domain, and let Q be a target predicate on D such that correctness of a classifier for Q forces closure-orbit agreement on D. Then D admits a correct classifier for Q if and only if Q has no orbit-gap witness inside D. Equivalently, restricting the domain avoids the no-go only by eliminating all orbit gaps of Q on that restricted domain.*

Proof sketch. The reverse implication is Corollary 4.13. For the forward implication, correctness-forced orbit agreement rules out any same-orbit disagreement inside D. The artifact proves this abstract principle directly, and also formalizes the optimizer-computation instance as a concrete compute-cost theorem. ■

This is the precise answer to the domain-restriction objection. Excluding the four named witness families is not enough by itself. Those families witness orbit gaps, but the logical boundary is orbit-gap freedom on the restricted closure-closed domain. If another orbit gap remains, the same impossibility theorem applies; if no orbit gap remains, the closure-hull construction yields a correct classifier on that domain.

5 The Realizability Barrier

One possible route to a finite tractability taxonomy would be to show that optimizer-induced quotients realize only a narrow subclass of equivalence relations, so that the frontier would emerge from realizability alone. In the unconstrained setting, this route fails.

Theorem 5.1[34] (Every Labeling Kernel Is Optimizer-Realizable). *Let $\phi : S \to T$ be any function. There exists a decision problem with action space T and state space S such that for every state $s \in S$,*

$$Opt(s) = \{\phi(s)\}.$$

Consequently, for all states $s, s' \in S$,

$$Opt(s) = Opt(s') \iff \phi(s) = \phi(s').$$

Equivalently, the decision quotient of the constructed problem is exactly the kernel partition of ϕ.

Proof. Take the action set to be T itself and define

$$U(a, s) = \begin{cases} 1 & \text{if } a = \phi(s), \\ 0 & \text{otherwise.} \end{cases}$$

Then the unique maximizer at state s is the designated action $\phi(s)$, so the optimizer set is exactly $\{\phi(s)\}$. Two states therefore have the same optimizer set if and only if they receive the same label under ϕ. ■

Theorem 5.1 is the key negative fact for the frontier program. It shows that optimizer realizability by itself is extremely permissive. If arbitrary label kernels are already optimizer-induced, then no finite taxonomy can come merely from asking which abstract quotients are realizable.

Proposition 5.2[35] (Every Equivalence Relation Is Optimizer-Realizable). *For every equivalence relation on the state space, there exists a decision problem whose decision quotient relation is exactly that equivalence relation. Moreover, the quotient object of the realizing problem is canonically equivalent to the corresponding setoid quotient.*

[33] : FR252-256 [34] : FR7-8 [35] : FR45-47

Proof. Let $\approx$ be an equivalence relation on S, and let $\pi : S \to S/\approx$ be the quotient map. Apply Theorem 5.1 to the labeling map π. The resulting decision problem satisfies

$$\mathrm{Opt}(s) = \mathrm{Opt}(s') \iff \pi(s) = \pi(s'),$$

and the right-hand side is exactly the original equivalence relation. Since the realizing quotient identifies states precisely by equality of quotient labels, its quotient object is canonically equivalent to the setoid quotient itself. ■

Proposition 5.2 is the maximal realizability statement available in the present framework. The labeling-kernel theorem is a special case. Thus the realizability barrier is not merely broad; at the level of abstract quotient relations, it is maximal.

Proposition 5.3[36] (The Realized Quotient Is Exactly the Label Range). *For every labeling map $\phi : S \to T$, the decision quotient of the realizing problem is canonically equivalent to the range of ϕ. In finite settings, the quotient cardinality is therefore exactly $|\mathrm{range}(\phi)|$.*

Proof. By Theorem 5.1, two states lie in the same decision class if and only if they receive the same label under ϕ. Therefore the quotient class of a state depends only on the value $\phi(s)$, and sending the class of s to $\phi(s)$ defines a well-defined map from the decision quotient to $\mathrm{range}(\phi)$. This map is surjective by definition of the range and injective because equality in the range means equality of labels, hence equality of quotient classes. ■

Proposition 5.3 strengthens Theorem 5.1. The realizing construction does not merely realize the induced equivalence relation abstractly; it realizes the quotient object itself with exactly the expected image size.

Corollary 5.4[37] (Realizability Alone Cannot Be the Frontier). *Any future optimizer-compatible tractability metatheorem must use more than quotient realizability alone. Additional restrictions have to enter through representation, coordinate structure, utility structure, closure properties of the family under study, or comparable algebraic invariants.*

This shifts the burden of the metatheorem. The hard question is not which quotients can arise in principle; almost all kernels can. The hard question is which *natural structural families of decision problems* force tractability or preserve hardness once coordinate structure and utility representation are taken seriously.

6 Necessary Conditions on Any Frontier Statement

The realizability barrier leaves a more precise frontier question. If quotient shape alone is too expressive, what should replace it as the organizing unit of the metatheorem?

At minimum, the tractability frontier should not classify arbitrary sets of decision problems. It should range over representation-level structural classes and it should ignore benign changes of presentation.

The right eventual condition list will almost certainly be stronger than this minimal principle. For example, a mature frontier theorem may also require closure under adding explicitly irrelevant coordinates, adjoining dummy actions, or other representation-preserving refinements. But action/state relabel invariance is the irreducible starting point.

[36] : FR39-40 [37] : FR7-8

Proposition 6.1[38] (Exact Certification Depends Only on the Decision Quotient Relation). *If two decision problems on the same state space induce the same decision-equivalence relation, then they induce the same exact-certification structure. In particular, they have the same sufficient coordinate sets, the same minimal sufficient coordinate sets, and the same relevant and irrelevant coordinates. Moreover, their quotient objects are canonically equivalent; in finite settings, those quotient objects therefore have the same cardinality.*

Proposition 6.1 is the canonical statement behind the frontier program. Exact certification is determined by the decision quotient relation itself. Equality of optimizer maps is only a sufficient route to this conclusion, because equal optimizer maps induce the same decision-equivalence relation.

Corollary 6.2[39] (Sufficiency Is Relation Refinement). *A coordinate set I is sufficient if and only if the coordinate-agreement relation induced by I refines the decision quotient relation.*

Corollary 6.2 is the cleanest relational form of exact certification. It removes utility language entirely: sufficiency asks whether one state relation is contained in another.

Corollary 6.3[40] (Relevance Is Failure of Erased Sufficiency). *A coordinate i is irrelevant if and only if deleting it leaves a sufficient coordinate set, namely* $\mathrm{univ} \setminus \{i\}$. *Equivalently, i is relevant if and only if that erased set is not sufficient.*

Corollary 6.3 makes the role of individual coordinates canonical: relevance is exactly the obstruction to retaining sufficiency after coordinate erasure.

Corollary 6.4[41] (Certification Statistics Factor Through the Quotient Relation). *Any statistic that is a function of the sufficient-set family and relevant-coordinate set is invariant under equality of the decision quotient relation. In particular, the number of sufficient coordinate sets, the number of minimal sufficient coordinate sets, and the number of relevant coordinates are quotient invariants.*

Corollary 6.4 records the next logical consequence of Proposition 6.1. Once exact certification is identified with quotient data, every certification statistic built from sufficient sets or relevant coordinates factors automatically through that quotient data as well.

Proposition 6.5[42] (Zero-Distortion Summaries Refine the Optimizer Quotient). *Let $\sigma : S \to C$ be a summary map. Suppose that*

$$\sigma(s) = \sigma(s') \implies Opt(s) = Opt(s') \qquad \text{for all } s, s' \in S.$$

Then each summary fiber is contained in a single optimizer-quotient class. Equivalently, σ factors through the optimizer quotient relation, and distinct optimizer classes require distinct summary symbols.

Proof. If $\sigma(s) = \sigma(s')$, the hypothesis gives $\mathrm{Opt}(s) = \mathrm{Opt}(s')$. So every σ-fiber is contained in one decision-equivalence class. This is exactly the statement that σ refines the optimizer quotient. Conversely, if two distinct optimizer classes shared a summary symbol, choosing representatives from those classes would violate the zero-distortion premise. ■

Proposition 6.5 explains why exact relevance is not merely one point on an approximation spectrum. Once distortion is required to be zero, lossless decision-preserving abstraction is already constrained by the optimizer quotient itself. Later propositions in this section sharpen the approximation boundary: approximate surrogate claims about relevance need explicit stability control relative to the exact optimizer sets, because uniform closeness alone does not preserve the exact decision boundary.

[38] : FR30-35, FR37, FR51 [39] : FR48 [40] : FR49-50 [41] : FR41-44 [42] : FR176-177

Corollary 6.6[43] (Constant-Optimizer Collapse Forces Trivial Certification). *If the optimizer set is constant across all states, then the empty coordinate set is sufficient and every coordinate is irrelevant.*

Corollary 6.6 isolates the first genuinely degenerate mechanism in the tractable basis. The certification problem disappears because there is no state dependence left to certify.

Proposition 6.7[44] (Explicit Enumeration Is Parameter-Dependent, Not Structural). *For every fixed exponent k, there exists a Boolean-cube family with state space size $2^n > n^k$. Thus explicit-state enumeration can outrun any fixed monomial bound in the ambient dimension parameter.*

Proposition 6.7 is weaker than a full polynomial lower-bound metatheorem, but it already captures the structural point needed here: tractability by brute-force state enumeration depends on a separate size parameter and does not arise from an optimizer-compatible mechanism like symmetry, low rank, or tree structure.

Proposition 6.8[45] (Positive Affine Utility Reparameterizations Are Invisible). *Statewise positive affine transformations of utility leave exact certification unchanged. In particular, replacing*

$$U(a, s) \qquad by \qquad \alpha(s) + \beta(s)U(a, s)$$

with $\beta(s) > 0$ for every state s does not change sufficient coordinate sets or relevance judgments.

Proposition 6.8 is another direct consequence of Proposition 6.1. It records a standard utility-theoretic invariance in the exact-certification language: only the induced decision quotient matters, not the particular positive affine scale used to encode utility.

Proposition 6.9[46] (Duplicate Actions and Duplicate States Preserve Certification). *Duplicating a single action without changing its utility profile preserves the decision quotient relation and hence preserves sufficiency and relevance. At the optimizer-set level, the only possible change is the addition of the duplicate action itself: original optimal actions remain optimal, and the duplicate is optimal exactly when the original action was. Duplicating a single state without changing its utility profile preserves the decision quotient up to the obvious quotient equivalence and also preserves sufficiency and relevance.*

Proposition 6.9 gives two more theorem-forced closure laws. These are not modeling conventions; they are consequences of the fact that exact certification factors through quotient data rather than through accidental multiplicity in the presentation.

Proposition 6.10[47] (Relabeling Invariance Is Forced by Exact Certification). *Action relabeling preserves optimizer equivalence and exact sufficiency. State relabeling does so as well once the coordinate structure is transported along the state bijection. Consequently, any structural tractability frontier for exact relevance certification must be closed under these relabelings.*

Proposition 6.10 is now best read as a corollary of Proposition 6.1. Bijective relabelings do not change the optimizer map up to the obvious identification, so they cannot change exact certification.

The second necessary ingredient is a genuine expressivity restriction. Call a class *kernel-universal* if, for every labeling map $\phi : S \to T$, it contains some decision problem whose optimizer quotient realizes exactly the kernel partition of ϕ. By Theorem 5.1, kernel universality is an extremely strong property.

The canonical finite invariant here is the size of the optimizer quotient, equivalently the number of distinct optimal-action sets. The quotient is identified with the range of the optimizer map, so quotient size is not auxiliary bookkeeping; it is intrinsic.

[43] : FR52-53 [44] : FR66-67 [45] : FR13-16 [46] : FR54-56, FR68-73 [47] : FR17-20

Proposition 6.11[48] (Quotient Size Is Unbounded Under Realizability). *For every $m \in \mathbb{N}$, there exists a finite decision problem whose optimizer quotient has size exactly m.*

Proof. Take $m = k+1$ and realize the identity labeling on $\mathrm{Fin}(k+1)$. Proposition 5.3 identifies the quotient object with the range of that identity labeling, which has cardinality $k+1$. Equivalently, Proposition 5.2 realizes the discrete equivalence relation on $\mathrm{Fin}(k+1)$. ■

A meaningful frontier statement cannot range over classes that are simultaneously relabel-invariant and kernel-universal, because quotient shape alone then becomes too expressive to isolate the boundary.

Proposition 6.12[49] (Invariance Alone Is Too Weak). *The universal class of all decision problems satisfies action relabeling invariance and state relabeling invariance, but it is kernel-universal. Consequently, relabeling invariance by itself is far too weak to isolate a tractability frontier.*

Proposition 6.12 separates two kinds of conditions that can otherwise be conflated.

1. **Benign invariance axioms** say the family should ignore arbitrary naming choices.
2. **Expressivity-limiting axioms** say the family should not realize arbitrary quotient structure wholesale.

Any genuine optimizer-compatible frontier theorem will need both. Without decision-quotient invariance and its immediate corollaries such as relabel invariance and positive affine invariance, the statement is not structural. Without an expressivity restriction excluding kernel universality or something comparably strong, the realizability theorem and Proposition 6.11 make the class too large for quotient shape alone to carry the boundary.

Proposition 6.13[50] (Independent Summary Frameworks Converge on the Same Structural Core). *Write $r = \mathrm{srank}(\mathcal{D})$ for the number of relevant coordinates of a Boolean decision problem, and let m be the number of optimizer-quotient classes. Then:*

1. *$m \le 2^r$, so the counting entropy $\log_2 m$ is at most r;*
2. *the diagonal $0/1$ relevance-support matrix has rank exactly r;*
3. *every zero-distortion decision-preserving summary requires at least m distinct summary symbols.*

Hence quotient entropy, Fisher-style support counting, and zero-distortion summary size are all controlled by the same support/quotient core.

Proof. If two states agree on all relevant coordinates, then they differ only on irrelevant ones. Changing those irrelevant coordinates one at a time cannot change the optimizer, so the relevant coordinates already determine the optimizer class. Therefore there are at most 2^r optimizer classes, one for each Boolean assignment on the relevant support, and $\log_2 m \le r$ follows.

For the second clause, the diagonal relevance-support matrix has a 1 exactly on relevant coordinates and a 0 elsewhere. Its rank is therefore the number of relevant coordinates, namely r.

The third clause is exactly Proposition 6.5: a zero-distortion summary must assign distinct symbols to distinct optimizer classes. ■

[48] : FR24, FR38 [49] : FR21-23 [50] : FR178-182

Proposition 6.13 does not identify entropy, support counting, and lossless summaries as identical notions. It shows that they already converge on the same structural content before any admissibility axiom is imposed. Closure-law invariance follows from this convergence: once several independent frameworks ignore surface presentation in favor of the support/quotient core, a structural tractability classifier should do the same. The remaining admissibility clauses serve as algorithmic and semantic guardrails.

From Necessary Conditions to an Admissibility Class

The closure properties above are theorem-forced by exact-certification semantics, but they are not enough on their own to support a meaningful impossibility theorem. If one allows unrestricted predicates, a semantic predicate can simply encode the target complexity class and trivialize the statement.

Proposition 6.14[51] (Unrestricted Predicates Trivialize Exact Characterization). *Without admissibility restrictions, there exist normalization predicates whose membership condition directly encodes the intended polynomial-time side of the landscape, so unrestricted collapse-impossibility claims fail.*

Proposition 6.14 is not a by-product of the proof method. It is the expected behavior of unrestricted semantic predicates. Therefore any meaningful frontier theorem must be parameterized by an admissibility class that simultaneously (i) respects theorem-forced invariances and (ii) blocks direct semantic encoding.

Proposition 6.15[52] (Closure Operations Preserve Exact Certification). *The theorem-forced closure operations preserve the exact-certification problem itself. More precisely:*

1. *action relabeling, coordinate relabeling, and statewise positive affine reparameterization preserve sufficient coordinate sets and relevant coordinates under the evident transport;*
2. *action duplication and state duplication preserve sufficient coordinate sets and relevant coordinates exactly;*
3. *binary irrelevant-coordinate extension preserves sufficiency after lifting coordinate sets, preserves relevance on the original coordinates, and makes the new coordinate irrelevant.*

In particular, each closure step induces the same exact-certification decision problem up to an explicit coordinate-set transport. For the irrelevant-coordinate case in the present binary-pairwise formalization, the encoding blowup is only constant-factor because the added coordinate is binary.

Proof sketch. Each clause is verified directly in the artifact at the level of exact-certification semantics. The relabeling and positive-affine cases preserve the decision-equivalence relation, hence preserve sufficiency and relevance. The duplication cases preserve decision-equivalence through the explicit quotient projections back to the original problem. The irrelevant-coordinate case is formalized as binary noise extension: $I \subseteq [d]$ is sufficient for the base slice if and only if its lift is sufficient after extension, the original coordinates remain exactly the relevant ones, and the new coordinate is irrelevant. Table 1 summarizes the theorem-backed transport statement. The encoding remarks in the right-hand column are direct from the concrete binary-pairwise representation and are not separate Lean claims. ■

Definition 6.16 (Admissible Normalization Predicate). A normalization predicate is *admissible* when it satisfies all of the following:

[51] : `FR145` [52] : `FR15-20`, `FR55-60`, `FR83-85`

Operation	Exact-certification transport	Encoding effect
Action/state relabeling	same sufficient sets and relevant coordinates after transport	relabeling only
Positive affine reparameterization	same sufficient sets and relevant coordinates	same arity, same action set; utility magnitudes rescaled
Action/state duplication	same sufficient sets and relevant coordinates	carrier duplication only
Binary irrelevant-coordinate extension	$I \leftrightarrow \mathrm{lift}(I)$, old relevance preserved, new coordinate irrelevant	arity increases by one binary coordinate

Table 1: Closure-operation transport for exact certification. The transport column is theorem-backed in Lean; the encoding-effect column records the direct representation-level effect of each operation in the binary-pairwise formalization.

1. polynomial-time checkability at the slice level;
2. invariance under the closure laws forced by exact certification (relabelings, positive affine reparameterizations, duplication operations, and binary irrelevant-coordinate extension);
3. structural extractability of the associated dependency graph;
4. bounded-pattern definability.

More explicitly, let $|U|$ denote the encoding size of a binary pairwise slice U, and let $X(U)$ be its canonical finite pairwise syntax. Polynomial-time checkability means that there exist constants $c, k \in \mathbb{N}$ and a decision procedure A such that

$$A(U) = 1 \iff Q(U), \qquad \mathrm{time}_A(U) \leq c(|U| + 1)^k + c.$$

Structural extractability means that whenever $Q(U)$ holds, the associated coordinate graph is obtained from syntax alone: there exists a uniform extractor

$$E : X \mapsto G_E(X)$$

on finite pairwise syntactic presentations such that the graph attached to U is exactly $G_E(X(U))$.

Bounded-pattern definability is likewise finite and explicit. There must exist uniform bounds on radius, neighborhood size, action alphabet size, and coefficient magnitude, together with finite lists of permitted and forbidden rooted local patterns, such that membership in Q is determined by the presence of one witness pattern or the absence of all forbidden patterns in the rooted interaction neighborhoods of $X(U)$. No unbounded global computation is hidden inside the definition.

Polynomial-time checkability is the minimal algorithmic requirement. A tractability characterization that cannot itself be recognized efficiently has little explanatory value: it would only relocate the computational difficulty from exact certification to the membership test for the classifier.

Closure-law invariance is the semantic requirement used in the orbit-witness arguments of Section 4. It is forced by the Block 6 invariance theorems, because exact certification itself is unchanged by relabelings, positive affine utility reparameterizations, duplication operations, and binary irrelevant-coordinate extensions.

Structural extractability distinguishes structural characterizations from purely semantic ones. Without such a condition, one can define predicates directly on optimizer behavior or on the solved certification instance itself, bypassing the stated aim of classifying tractability by representation-level structure.

Bounded-pattern definability is the locality guardrail. Its role is to exclude predicates whose membership test secretly performs an unbounded global computation under a structural name. In particular, it prevents one from reintroducing oracle power through an arbitrarily large finite schema.

Theorem 6.17 (Tractability Classifiers Are Forced to Be Closure-Invariant). *Let C be any procedure assigning to each representation in a closure-closed domain Γ a verdict in $\{\text{tractable}, \text{intractable}\}$, and suppose that C correctly predicts whether exact certification is polynomial-time decidable on the underlying problem. Then C must agree on representations within the same closure orbit. Equivalently, every correct tractability classifier on a closure-closed domain is closure-invariant on that domain, regardless of whether its internal features are themselves invariant.*

Proof sketch. This paper theorem is formalized in `AdmissibleCharacterization.lean` as the composition of the generic correctness-on-domain orbit-agreement theorem with the closure-invariance inheritance theorem. If $U, V \in \Gamma$ lie in the same closure orbit, Proposition 6.15 shows that they induce the same exact-certification problem up to the explicit coordinate-set transport attached to the relevant closure step. The theorem uses the semantic notion of correctness stated above: tractability is attached to that underlying exact-certification problem, not to arbitrary relabelings or duplicate-label presentations of the same problem. Correctness of C therefore forces the same verdict on both. ■

Theorem 6.18[53] (Compute-Cost Version: Optimizer Computation). *Let C be any procedure assigning to each representation in a closure-closed domain Γ a verdict in $\{\text{tractable}, \text{intractable}\}$, and suppose that C correctly predicts whether optimizer computation is polynomial-time solvable on the underlying problem. Then C must agree on representations within the same closure orbit.*

Proof sketch. This is the optimizer-computation instance of the same correctness-on-domain theorem. The compute-cost transport layer in the artifact shows that each closure step preserves polynomial-time solvability of optimizer computation under its explicit state and output transports. Correctness of C therefore forces the same verdict on closure-equivalent representations. ■

Corollary 6.19[54] (Compute-Cost Version: Canonical Payload and Search Tasks). *The same closure-orbit agreement conclusion holds when C correctly predicts polynomial-time solvability of deterministic optimizer-set payload output. It also holds when C correctly predicts polynomial-time solvability of admissible-output search on optimizer-set semantics.*

Proof sketch. Apply the same correctness-on-domain theorem to the two canonical compute-cost families built from optimizer-set payload output and optimizer-set admissible-output search. ■

Thus the forced-invariance argument applies not only to exact-certification predicates, but also to all three canonical output-production tasks treated in the compute-cost layer.[55]

Corollary 6.20[56] (Compute-Cost Version: External Output Objects). *Let X be an external output class whose admissibility relation is preserved under the closure laws by pullback on states and*

[53] : `FR197-199`, `FR245-249` [54] : `FR248` [55] : `FR245-249`, `FR259-262` [56] : `FR259-262`

identity on outputs. Then any correct classifier for polynomial-time search over admissible X-outputs on a closure-closed domain must agree on closure orbits, and the same orbit-gap no-go applies.[57]

Proof sketch. This paper-level corollary aggregates the generic identity-output closure theorem, the bridge to transported outputs, and the named external-output instances formalized in Lean. The generic external-output compute-cost layer transports only the input state; the output object itself is unchanged. Once the admissibility relation respects those state pullbacks, the same correctness-on-domain theorem applies. ■

Corollary 6.21 (Compute-Cost Version: Representation-Relative Output Objects). *Let X_U be a representation-relative output object whose type and admissibility relation may vary with the representation U, and suppose each closure witness carries an explicit output transport preserving admissibility. Then any correct classifier for polynomial-time search over admissible X_U-outputs on a closure-closed domain must agree on closure orbits, and the same orbit-gap no-go applies. The artifact includes named instances for representation-relative hypotheses, estimators, policies, and randomized procedures.*[58]

Proof sketch. This paper-level corollary aggregates the generic transported-output closure theorem with the named representation-relative hypothesis, estimator, policy, and randomized-procedure instances. The generic transported-output compute-cost layer packages the witness-by-witness output maps together with the corresponding admissibility preservation laws, and the same correctness-on-domain theorem then applies. ■

This answers the natural objection that Section 4 excludes only classifiers that advertise closure invariance as an explicit axiom. Correctness already forces the same orbit agreement. The closure laws therefore supply the representation-level congruence needed for the orbit-gap program: once two slices are related by these theorem-forced presentation equivalences, any correct tractability classifier must identify them. Theorem 4.9 therefore applies to any correct classifier on a closure-closed domain once the target notion is tractability itself: if the classifier separates same-orbit witnesses, it is simply wrong on at least one of them.

Proposition 6.22[59] (The Admissibility Layer Is Nonempty). *Definition 6.16 is not vacuous. There exist admissible normalization predicates; in fact both the constant-true and constant-false slice predicates are admissible.*

Proof sketch. Both predicates are decidable in constant time and are automatically invariant under all closure laws. Their graph extractors are fixed graphs independent of the input slice. For bounded-pattern definability, use a single impossible local pattern: a radius-zero pattern with two distinct vertices cannot occur in any slice. Taking that pattern as forbidden yields the constant-true predicate, and taking it as a required witness yields the constant-false predicate. ■

Thus the impossibility theorem is not an emptiness statement about Definition 6.16. The admissibility layer already contains explicit predicates; the point of the no-go theorem is that correctness across the obstruction families forces a contradiction once closure-orbit agreement is respected.

Proposition 6.23[60] (Bounded Distinct Action Profiles Compress to Bounded Actions). *For a binary pairwise slice U, let $d(U)$ be the number of distinct action utility profiles. Then there exists a compressed slice U^{prof} with exactly $d(U)$ actions such that a coordinate set is sufficient for U if and only if it is sufficient for U^{prof}, and a coordinate is relevant for U if and only if it is relevant for*

[57] : FR316-328 [58] : FR326-336 [59] : FR200 [60] : FR205-209

U^{prof}. *Consequently, if $d(U) \le k$, the bounded-actions tractability theorem applies to U^{prof}, so the bounded-distinct-profile subcase is polynomial on the exact-certification side.*

Proof sketch. Replace each action by its full utility profile over states and quotient the action set by profile equality. Duplicated actions collapse to a single profile action, but exact certification is unchanged because optimizer equality depends only on which profiles are optimal, not on how many labels realize a given profile. The compressed slice has exactly $d(U)$ actions by construction, and the artifact proves equality of the sufficient-coordinate and relevant-coordinate predicates between U and U^{prof}. The existing bounded-actions polynomial-time theorem then applies to the compressed slice. ■

Proposition 6.23 isolates a genuine positive tractable subcase by compression to bounded actions. It is stated as a compression theorem rather than direct admissibility membership because Definition 6.16 fixes a bounded-pattern action alphabet in advance. The proposition still shows that nontrivial positive classification work is available through closure-law-respecting reduction inside the present framework.

Proposition 6.24[61] (Bounded-Pattern Predicates Stabilize Above a Finite Action Bound). *For every bounded-pattern definable predicate Q, there exists a finite bound B such that Q is constant on all slices with more than B actions. Equivalently, once the action alphabet exceeds the action bound built into the defining pattern scheme, the scheme can no longer distinguish between such slices.*

Proof sketch. In a bounded-pattern scheme every witness and forbidden pattern has action alphabet size at most a fixed bound B. If a slice has more than B actions, then no listed local pattern can occur in it, because occurrence requires an action-set equivalence between the pattern and the slice. The witness branch of the scheme is therefore impossible, while the forbidden branch is determined only by whether the forbidden list is empty. Hence the predicate is constant above the fixed action bound. ■

This is why Proposition 6.23 is stated as a compression theorem rather than as direct admissibility membership for the distinct-profile predicate. Unbounded profile counting does not naturally fit a definition whose bounded-pattern clause fixes a finite action alphabet in advance.

Proposition 6.25[62] (Deterministic Payload Transfer). *Let S be a coordinate space, let $\phi : S \to T$ be any deterministic payload map into a finite label type, and let D_ϕ be the induced decision problem with action space T and utility*

$$U(a, s) = \begin{cases} 1, & a = \phi(s), \\ 0, & a \neq \phi(s). \end{cases}$$

Then every coordinate set is sufficient for ϕ if and only if it is sufficient for D_ϕ, and every coordinate is relevant for ϕ if and only if it is relevant for D_ϕ. Equivalently, exact feature sufficiency and relevance for any deterministic payload reduce definitionally to exact relevance certification for an induced decision problem.

Proof sketch. For the induced decision problem, the optimizer at state s is the singleton $\{\phi(s)\}$. Hence two states have the same optimizer if and only if they carry the same payload value. The sufficiency and relevance predicates are therefore literally the same coordinate conditions on S in both formulations. ■

This is a theorem of exact semantic equivalence. Deterministic payload sufficiency and relevance are already exact relevance certification for the induced decision problem.

[61] : FR214-216 [62] : FR217-222

Corollary 6.26[63] (Boolean Payload Transfer). *Let $\phi : S \to \{0,1\}$ be any Boolean payload. Then exact feature sufficiency and relevance for ϕ reduce definitionally to exact relevance certification for the induced decision problem.*

Proof sketch. This is Proposition 6.25 specialized to the two-label codomain $\{0,1\}$. ■

Corollary 6.27[64] (Predicate Transfer). *Let $P(s)$ be any exact yes/no correctness predicate on the state space with decidable truth value. Writing its truth value as a Boolean payload, exact feature sufficiency and relevance for P reduce definitionally to exact relevance certification for the induced decision problem.*

Proof sketch. Encode P by its Boolean truth-value map and apply Corollary 6.26. ■

Corollary 6.28[65] (Nonempty Set-Valued Payload Transfer). *Let $F(s) \subseteq A$ be a nonempty feasible-action set at each state of a coordinate space, and equip allowed actions with utility u_{allowed} and blocked actions with utility u_{blocked} where $u_{\text{blocked}} < u_{\text{allowed}}$. Then exact feature sufficiency and relevance for the set-valued payload F are equivalent to exact relevance certification for the induced decision problem whose optimal set at state s is exactly $F(s)$.*

Proof sketch. This is the set-valued analogue of Proposition 6.25. The strict utility gap makes the optimizer set coincide with the feasible-action set at every state, so the same coordinate conditions define sufficiency and relevance on both sides. ■

Corollary 6.28 covers total search semantics stated as nonempty feasible-witness sets. Deterministic payloads are the singleton special case.

Corollary 6.29[66] (Arbitrary Set-Valued Payload Transfer via Failure Tokens). *Let $F(s) \subseteq A$ be any set-valued payload, possibly empty. Form the totalized payload on* Option(A) *by adjoining a distinguished failure token whenever $F(s) = \varnothing$. Then exact feature sufficiency and relevance for F are equivalent to exact relevance certification for the induced decision problem on the totalized payload.*

Proof sketch. Map each admissible output $a \in F(s)$ to `some`(a), and use `none` exactly when $F(s)$ is empty. Equality of the original set-valued payloads is then equivalent to equality of the totalized nonempty payloads, so Corollary 6.28 applies after this totalization step. ■

Corollary 6.29 removes the empty-fiber caveat for search-style semantics. Set-valued payloads with failure states transfer without choosing a deterministic witness selector.

Corollary 6.30[67] (Arbitrary Exact Output Semantics Transfer). *Let $R(s,a)$ be any state-indexed admissible-output relation. Then exact feature sufficiency and relevance for the induced output semantics reduce to exact relevance certification for a decision problem obtained by totalizing the admissible-output sets with a failure token and assigning a strict allowed-versus-blocked utility gap.*

Proof sketch. Write $F(s) = \{a : R(s,a)\}$. This is exactly the arbitrary set-valued case of Corollary 6.29. The relation form is only a notational repackaging of the same theorem. ■

Proposition 6.31[68] (Exactness Means Exact Agreement With Validity). *Let $V(s,a)$ be any state-indexed validity relation. Then the exact relevance profile induced by V is realized by exact relevance certification for the induced decision problem. In particular, exactness refers to exact agreement with V itself, not to zero-error determinism or the absence of approximation, randomization, statistical thresholds, or failure states in the specification.*

[63] : `FR340` [64] : `FR341` [65] : `FR223-225` [66] : `FR226-228` [67] : `FR229-231` [68] : `FR346`

Proof sketch. Corollary 6.30 identifies the sufficient-coordinate family and relevant-coordinate set of the validity semantics with those of the induced decision problem. The artifact packages the resulting profile equality directly. ■

This is the sense in which approximation, statistical, and randomized semantics remain exact: the specification written into V may be approximate or randomized, but agreement with V is exact.

Rigorous specification, in the exact sense relevant here, is the determination of such a validity relation. If a purported specification does not determine which outputs are valid at each state, objective correctness is not yet fixed, so the exact semantic questions studied here are not well-posed for it. Formal methods do not add this structure from outside; they make explicit the structure that rigor already requires. The accompanying Lean artifact packages this semantic floor directly as an `ExactCorrectnessSpecification`. Different coordinate presentations may make quotient recovery trivial or rich, but they do not change the canonical quotient carried by the specification.

Proposition 6.32[69] (Sufficiency Is Relation Refinement). *Let $R(s,a)$ be any exact correctness relation, write $\mathrm{Adm}_R(s) = \{a : R(s,a)\}$, and define*

$$s \sim_R s' \quad \Longleftrightarrow \quad \mathrm{Adm}_R(s) = \mathrm{Adm}_R(s').$$

Then a coordinate set I is sufficient if and only if agreement on I forces $s \sim_R s'$.

Proof sketch. This is the exact-semantics specialization of sufficiency as relation refinement. The relation being refined is now the admissible-output equivalence relation itself. ■

Corollary 6.33[70] (Relevance Is Erased Failure of Refinement). *Let $R(s,a)$ be any exact correctness relation. A coordinate i is relevant if and only if agreement on all coordinates except i does not force equality of admissible-output classes. Equivalently, the full coordinate set with i erased is not sufficient.*

Proof sketch. Erase coordinate i and apply Proposition 6.32 to the remaining agreement relation. ■

Theorem 6.34[71] (Exact Semantics Quotient Universality). *Let $R(s,a)$ be any exact correctness condition, write $\mathrm{Adm}_R(s) = \{a : R(s,a)\}$, and define*

$$s \sim_R s' \quad \Longleftrightarrow \quad \mathrm{Adm}_R(s) = \mathrm{Adm}_R(s').$$

Then $\sim_R$ is the canonical semantic object of the problem. Exact relevance certification for the induced decision problem realizes exactly the same sufficient-coordinate family and relevant-coordinate set; a coordinate set is sufficient exactly when it recovers the $\sim_R$-classes; and a coordinate is relevant exactly when erasing it destroys that recovery.

Proof sketch. The Lean artifact packages this paper theorem as `exactSemanticsQuotient_universal_characterization`, built from the exact-profile realization theorem together with the sufficiency-as-refinement and relevance-as-erased-failure lemmas. Corollary 6.30 reduces the exact semantics to exact relevance certification for the induced decision problem. Proposition 6.32 identifies sufficiency with recovery of the admissible-output classes, and Corollary 6.33 identifies relevance with failure of that recovery after erasure. ■

[69] : `FR342` [70] : `FR343` [71] : `FR345`

Proposition 6.35[72] (Canonical Exact Relevance Profile). *Every state-indexed admissible-output relation induces a canonical exact relevance profile, consisting of its sufficient-coordinate family and relevant-coordinate set. This profile is realized by exact relevance certification for the induced decision problem, and it depends only on admissible-output equivalence.*

Proof sketch. Corollary 6.30 identifies the sufficient-coordinate family and relevant-coordinate set of the semantics with those of the induced exact-certification decision problem. The quotient-level theorems then show that these two objects depend only on admissible-output equivalence. ■

Proposition 6.36[73] (Every Exact Specification Admits a Coordinate Presentation). *Every exact output specification admits a coordinate presentation. In the weakest form, one may take the entire state as a single coordinate. Under this presentation, the exact relevance profile of the specification is realized by exact relevance certification for the induced decision problem.*

Proof sketch. Use a one-coordinate presentation whose unique coordinate is the state itself. Agreement on that coordinate is equality of states, so the coordinate presentation is exact. The artifact then identifies the resulting exact relevance profile with the one carried by the induced exact-certification decision problem. ■

Proposition 6.37[74] (Finite Exact Specifications Admit Coordinate Presentations). *Every finite-state exact output specification admits a coordinate presentation. Concretely, if the state space is finite, one may index coordinates by states and use state-indicator bits as the coordinate map. Under this presentation, the exact relevance profile of the specification is realized by exact relevance certification for the induced decision problem. This strengthens Proposition 6.36 by replacing the trivial one-coordinate presentation with a finite Boolean one.*

Proof sketch. For a finite state space, assign one Boolean coordinate to each state and record whether the current state is that state. Agreement on all coordinates is then equality of states, so the presentation is exact. The artifact formalizes this indicator-coordinate presentation and shows that the resulting exact relevance profile is exactly the one carried by the induced decision problem. ■

Proposition 6.38[75] (Countable Exact Specifications Admit Countable Boolean Presentations). *Every exact output specification on an encodable state space admits a countable Boolean presentation. If the state space is countable and carries the discrete measurable or discrete topological structure, this presentation can be chosen measurable or continuous, respectively.*

Proof sketch. Encode each state by a natural number and use its binary digits as Boolean coordinates indexed by $\mathbb{N}$. Equality of all bits is equality of encodings, hence equality of states. In the discrete measurable and discrete topological settings, each coordinate map is measurable or continuous automatically. ■

Proposition 6.39[76] (Finite Exact Specifications Admit Low-Dimensional Boolean Presentations). *Every finite-state exact output specification admits a Boolean coordinate presentation using only* $\mathrm{size}(|S|)$ *coordinates, where* $\mathrm{size}(|S|)$ *is the bit-length of the state-space cardinality. Under this presentation, the exact relevance profile is again realized by exact relevance certification for the induced decision problem.*

[72] : FR302-303 [73] : FR307-308 [74] : FR304-306 [75] : FR311-313 [76] : FR314-315

Proof sketch. Index states by $0, \ldots, |S| - 1$ and use the binary digits of the index as Boolean coordinates. Since every index is less than $2^{\text{size}(|S|)}$, equality of those finitely many bits determines the state. The artifact formalizes this low-dimensional Boolean presentation and identifies the induced exact relevance profile with exact relevance certification. ■

Every exact correctness claim admits a coordinate presentation, so the question of which coordinates matter is always a quotient-recovery problem for admissible-output classes. This includes decision, counting, search, approximation guarantees, PAC/regret/risk guarantees, anytime or finite-horizon guarantees, randomized-output guarantees, distributional specifications, and any other specification stated by the outputs that count as correct at each state.

Proposition 6.36 gives the universal one-coordinate realization, while Propositions 6.37–6.39 show that the same semantics can also admit finite or countable Boolean realizations. Vacuous and informative encodings are therefore different presentations of one semantic object, not different semantics. For example, SAT appears through its exact yes/no validity predicate, sorting through the set of correctly sorted outputs for each input, and fixed approximation or PAC guarantees through the outputs meeting the stated threshold.

Corollary 6.40[77] (Statistical Guarantee Semantics Transfer). *Let $R(s, a)$ specify the admissible outputs for a PAC guarantee, regret guarantee, statistical risk guarantee, anytime guarantee, or finite-horizon guarantee at state s. Then exact feature sufficiency and relevance for that guarantee semantics reduce to exact relevance certification for the induced decision problem.*

Proof sketch. Each case is a named specialization of Corollary 6.30. The output object may be a hypothesis, learner, estimator, or policy; correctness still means exact agreement with the admissible-output relation specifying the guarantee, not absence of statistical error thresholds in that guarantee. ■

Corollary 6.41[78] (Randomized-Output Guarantee Semantics Transfer). *Let $R(s, a)$ specify the admissible randomized outputs for a state s, where a may itself be a distribution, kernel, randomized estimator, or randomized policy. Then exact feature sufficiency and relevance for that randomized-output semantics reduce to exact relevance certification for the induced decision problem.*

Proof sketch. This is again a named specialization of Corollary 6.30. Randomization changes the output object, not the meaning of correctness: correctness still means exact agreement with the admissible-output relation on those randomized outputs, not deterministic singleton output semantics. ■

Corollary 6.42 (Randomized-Output Guarantee Semantics Inherit the Closure-Orbit Consequences). *Once a randomized-output guarantee is written as a state-indexed admissible-output relation, the same closure-orbit agreement and no-go theorems apply to classifiers for that semantics. In particular, Theorem 6.17, Corollary 4.11, and Corollary 4.14 apply unchanged after the transfer of Corollary 6.41.*

Proof sketch. This is the randomized-output instance of the same transfer-plus-application pattern used above. Corollary 6.41 reduces the semantics to exact relevance certification for the induced decision problem, and the dedicated randomized application theorems then import the same closure-orbit agreement and orbit-gap consequences. ■

[77] : FR268-282 [78] : FR291-293

Corollary 6.43[79] (Statistical Guarantee Semantics Inherit the Closure-Orbit Consequences). *Once a PAC, regret, statistical-risk, anytime, or finite-horizon guarantee is written as a state-indexed admissible-output relation, the same closure-orbit agreement and no-go theorems apply to classifiers for that semantics. In particular, Theorem 6.17, Corollary 4.11, and Corollary 4.14 apply unchanged after the transfer of Corollary 6.40.*

Proof sketch. Corollary 6.40 reduces each named statistical guarantee semantics to exact relevance certification for the induced decision problem. The artifact packages the next step as a generic transferred-semantics application principle: any semantics whose tractability predicate is definitionally equivalent to a closure-law-invariant target inherits the same closure-orbit agreement and orbit-gap consequences. Statistical guarantee semantics are named instances of that principle. They also inherit the same exact-semantics quotient invariance, because their sufficient-coordinate family and relevant-coordinate set depend only on admissible-output equivalence. ■

Corollary 6.44[80] (Exact Approximation Semantics Transfer). *Let $R(s, a)$ specify the admissible outputs for an exact approximation specification at state s, for example the set of all outputs meeting a fixed approximation guarantee. Then exact feature sufficiency and relevance for that approximation semantics reduce to exact relevance certification for the induced decision problem.*

Proof sketch. This is the approximation-specialized form of Corollary 6.30. Approximation enters only through the specification: once the admissible-output relation says which outputs meet the guarantee, correctness with respect to that approximation specification is exact agreement with that relation, not zero approximation error. ■

Proposition 6.45[81] (Approximate Relevance and Sufficiency Claims Need Explicit Stability Control). *Let D and D' be decision problems on the same coordinate space, and suppose their utilities are uniformly δ-close. If coordinate i has a relevance witness in D given by states s, s' with distinct strict optimal actions at the two witness states, and if the strict utility gap at each witness state exceeds 2δ, then i is also relevant in D'. Likewise, if a coordinate set I has a non-sufficiency witness in D given by states s, s' that agree on I but have distinct strict optimal actions, and if the strict utility gap at each witness state exceeds 2δ, then I is still not sufficient in D'.*

Proof sketch. The existing uniform-approximation theorem preserves the optimizer set at any state whose strict utility gap dominates the approximation error. Applying that theorem to both witness states keeps the two singleton optimizer sets distinct. In the relevance case this preserves the same relevance witness. In the sufficiency case the same state pair still agrees on I but still has different optimizer sets, so it remains a non-sufficiency witness. ■

Corollary 6.46[82] (Arbitrarily Small Uniform Perturbations Can Flip Relevance). *For every $\varepsilon > 0$, there exist two decision problems on the same one-coordinate Boolean state space that are uniformly ε-close, yet the unique coordinate is relevant in one problem and irrelevant in the other.*

Proof sketch. The artifact contains an explicit two-action, one-coordinate construction. In the first problem the optimizer tracks the Boolean state, so the unique coordinate is relevant. In the second problem all actions are tied at every state, so the coordinate is irrelevant. The two utility functions differ uniformly by at most ε. ■

[79] : FR291-296 [80] : FR232-234 [81] : FR263, FR266 [82] : FR264

Corollary 6.47[83] (Arbitrarily Small Uniform Perturbations Can Flip Sufficiency). *For every $\varepsilon > 0$, there exist two decision problems on the same one-coordinate Boolean state space that are uniformly ε-close, yet the empty coordinate set is sufficient in one problem and not sufficient in the other.*

Proof sketch. The same one-coordinate construction separates the constant-optimizer case from the state-tracking case. In the flat problem the empty set is sufficient because the optimizer set is constant, while in the state-tracking problem the empty set is not sufficient because the two Boolean states induce different optimizer sets. The two utility functions are still uniformly ε-close. ■

Proposition 6.48[84] (Global Approximation Stability Under Uniform Strict Gaps). *Let D and D' be decision problems on the same coordinate space, and suppose their utilities are uniformly δ-close. If every state of D has a strict optimal action whose utility gap exceeds 2δ, then D and D' have the same optimizer quotient, the same sufficient-coordinate family, the same relevant-coordinate set, and the same minimal sufficient sets.*

Proof sketch. The uniform-approximation theorem preserves the optimizer set at each state once the local strict utility gap dominates the approximation error. Applying this statewise yields equality of optimizer sets for all states. The optimizer quotient, sufficient-coordinate family, relevant-coordinate set, and minimal sufficient sets therefore coincide. ■

Thus closeness alone does not justify an approximate relevance or sufficiency claim. Any ε-based surrogate claim needs an explicit reduction to the exact optimizer sets, for example by a witness-gap bound as in Proposition 6.45, because otherwise arbitrarily small perturbations can change the exact judgment.

Proposition 6.49[85] (Exact Semantic Claims Depend Only on Admissible-Output Equivalence). *Let R and R' be two exact output semantics on the same state space. If they induce the same equality relation on admissible-output sets, then they induce the same sufficient coordinate sets and the same relevant coordinates. Exact semantic claims about sufficiency and relevance therefore depend only on the induced admissible-output equivalence relation.*

Proof sketch. The transfer theorems reduce both semantics to exact relevance certification for induced decision problems. If the admissible-output equality relation is the same, the induced sufficiency and relevance structures are the same as well. The artifact proves equality of the sufficient-set family and of the relevant-coordinate set under this hypothesis. ■

Corollary 6.50[86] (Every State Equivalence Relation Is Realizable as Exact Output Semantics). *For every equivalence relation on the state space, there exists an exact output semantics whose admissible-output equality relation is exactly that equivalence relation.*

Proof sketch. Take the output space to be the quotient by the given equivalence relation and assign to each state its own quotient class as a singleton admissible-output set. Equality of those singleton output sets is then exactly the original equivalence relation. ■

Proposition 6.51[87] (Semantically Extensional Claims Factor Through the Exact-Semantics Quotient). *Let C be any claim about exact output semantics that depends only on admissible-output equivalence. Then C factors through the quotient of output semantics by admissible-output equivalence. In particular, once a semantic claim is exact and extensional, the exact-semantics quotient is the object of the claim.*

[83] : FR267 [84] : FR288-301 [85] : FR235-236 [86] : FR237, FR240 [87] : FR241-244

Proof sketch. This is the quotient-factorization theorem for semantically extensional maps, specialized to `Prop`-valued claims. If C takes the same truth value on any two semantics with the same admissible-output equivalence, then C descends to the quotient by that equivalence relation. ■

Closure-law invariance (clause 2) is not an arbitrary stylistic restriction. Theorem 6.17 shows that correctness already forces closure-orbit agreement, and Theorem 4.9 shows that this theorem-forced congruence is the only load-bearing hypothesis in the no-go theorem. The remaining clauses are the minimal guardrails against the failure modes identified in Proposition 6.14; the specific guardrail formulation of clauses 1, 3, and 4 is one natural choice but is not load-bearing.

7 Related Work

Rough Sets, Feature Selection, and Exact Relevance

At the static level, rough-set reduct theory [16, 26] is the clearest classical comparison point. That literature studies which attributes can be deleted while preserving decision distinctions. Exact static sufficiency is a reduct condition for the induced decision table. The question here is different: which tractable mechanisms admit a finite characterization?

A separate literature studies feature selection, variable importance, and model explanation in machine learning and AI, including wrapper and filter methods [12], general surveys [10], and attribution methods based on Shapley-style decompositions [13]. These works matter here by motivation rather than by technique. They typically optimize predictive performance, explanatory salience, or approximation quality under data-dependent criteria. Exact relevance certification instead asks for a semantic preservation property: which coordinates are necessary to preserve the outputs that a correctness condition counts as admissible. Section 6 proves that arbitrary exact admissible-output semantics reduce to this same quotient-recovery problem. This includes decision, counting, search, approximation guarantees, PAC/regret/risk guarantees, finite-horizon or anytime guarantees, randomized-output guarantees, and distributional specifications written as admissible-output relations.

Decision-theoretic informativeness and value of information, beginning with Blackwell's comparison of experiments and subsequent decision-theoretic treatments [3, 19, 11], provide a second comparison point. Both settings ask when one representation retains all decision-relevant distinctions of another. The difference here is the combinatorial complexity of certifying exact preservation under coordinate deletion.

Information-Theoretic and Transport Viewpoints

The zero-distortion discussion belongs to the classical information-theoretic tradition of exact distinguishability and lossless coding boundaries, going back to Shannon and standard modern treatments of source coding and rate-distortion theory [21, 22, 23, 5]. Fisher-style support counting supplies another classical comparison lens on relevant support [7]. These viewpoints are used only structurally. There is no general coding or statistical estimation problem here. Zero distortion, entropy, and support counting appear only to show that several independent summary frameworks are already constrained by the same optimizer-quotient core. The Wasserstein language in Section 4 plays the same role: a qualitative description of quotient branching rather than a full transport-theoretic model.

Backdoors, Tractable Islands, and Dichotomy Programs

The nearest complexity-theoretic analogue is backdoor tractability for constraint satisfaction [27, 2, 8]. There, a small structural set exposes membership in a tractable class even when the ambient problem is hard. The comparison is direct: exact relevance certification also asks which coordinates matter, and polynomial-time behavior emerges when the source of hardness is restricted by structure. Bessiere et al. [2] make especially clear that exploiting a known tractable structure and discovering the responsible structure are different algorithmic tasks; later parameterized work sharpened that distinction. This is closely aligned with the present paper's separation between positive tractable mechanisms and the harder meta-level problem of recognizing a correct frontier classifier.

The broader analogy is the dichotomy tradition for satisfiability and constraint satisfaction, from Schaefer's Boolean dichotomy theorem [20] to the finite-domain CSP dichotomy theorems of Bulatov and Zhuk [4, 28]. The expectation that a clean tractability boundary should exist was articulated in the Feder–Vardi program [6], and later algebraic work of Barto and Kozik clarified why finite structural boundaries are plausible in that setting [1]. The point here is narrower: for output specifications of any kind, quotient realizability and closure-law invariance defeat the most direct admissible classifier, both at the feature-sufficiency layer and at the compute-cost layer for optimizer computation, canonical payload/search tasks, and theorem-backed external or transported outputs. This is compatible with the CSP dichotomy program and helps explain its form: once admissible closure-invariant classifiers are ruled out, positive frontier theorems must use stronger algebraic structure outside Definition 6.16. The theorem turns an empirical lesson of the CSP literature into a structural one: Bulatov–Zhuk style frontier theorems had to use stronger algebraic invariants, because anything confined to the direct closure-invariant structural regime was already impossible. The open problem is to identify an analogue of that stronger structure for the present setting.

The quotient viewpoint also touches the literature on exact abstraction and aggregation in stochastic control, Markov decision processes, and reinforcement learning [15, 9, 17]. Those works study when states may be aggregated while preserving value or policy structure. Our setting is narrower: we ask for exact preservation of optimizer classes under coordinate-hiding maps rather than arbitrary state abstractions. Still, the common theme is that semantic invariants of decision behavior should survive presentation-level simplification.

Meta-Impossibility Traditions

The closest analogy is Rice-style impossibility. Rice's theorem [18] shows that nontrivial extensional properties of partial recursive functions are undecidable, with extensionality forced by what "semantic" means rather than chosen as an axiom. Refinements such as the Myhill–Shepherdson theorem and the Rice–Shapiro theorem extend this pattern to broader structured semantic domains [14, 24]. The present theorem has the same structure with a different forced invariance. Proposition 6.1 shows that exact certification depends only on quotient data, and Theorem 6.17 lifts this to tractability classification: any correct tractability classifier, for any output specification written as an admissible-output relation, must agree on closure-equivalent representations. The no-go then depends on this forced invariance alone (Theorem 4.9), with Corollary 4.11 giving the resulting impossibility for correct classifiers. The remaining admissibility clauses play the role of algorithmic and structural guardrails that exclude classifiers admissible in name only, but they do not enter the contradiction.

8 Conclusion

Any rigorously specified computational problem determines a state-indexed admissible-output relation R, the only state distinctions that matter are the admissible-output equivalence classes $s \sim_R s' \iff \mathrm{Adm}_R(s) = \mathrm{Adm}_R(s')$, and every exact correctness claim reduces to the same quotient-recovery problem. Exact relevance certification asks which coordinates recover those classes. Section 6 proves that exact predicates, decision problems, counting, search, approximation, PAC/regret/risk, finite-horizon or anytime, randomized-output, and distributional guarantees all reduce to that same quotient-recovery problem. Every state equivalence relation is realizable at this semantic layer, every exact claim admits a coordinate presentation, and semantically extensional claims factor through the same quotient. Exact correctness is not structure imposed by formal methods; it is the structure a specification must carry for objective correctness, and hence tractability of the underlying problem, to be well-posed at all. The Lean development also formalizes the compute-cost closure-orbit layer: optimizer computation, deterministic optimizer-set payload output, admissible-output search, external output objects, and representation-relative transported outputs such as policies and randomized procedures all admit theorem-backed closure transport, and correctness of a polynomial-time classifier for those tasks forces the same closure-orbit agreement.

On the positive side, the tractable landscape factors into primitive mechanisms, lifts, and degeneracies. The eight-mechanism basis is the explicit inventory of the current positive landscape, not the universal theorem. Proposition 6.23 shows that this positive side remains substantive at the representation level: bounded distinct action profiles compress to bounded-action slices without changing exact certification, so the existing bounded-actions theorem already yields a genuine structural classifier on that subcase. On the negative side, quotient realizability is maximal, closure-law invariance is forced by correctness, and the orbit-gap mechanism defeats finite admissible classifiers across the four obstruction families. The same pair-targeted action-independent affine transport yields the admissible-collapse theorem and the concrete compute-cost no-go for polynomial-time optimizer computation. Domain restriction changes nothing in kind: it helps only by removing all remaining orbit gaps on the restricted closure-closed domain.

The open problem is therefore sharp. Simple finite lists, unconstrained quotient predicates, and admissible closure-invariant classifiers of the present kind cannot characterize the boundary. Any successful frontier theorem must use stronger representation-sensitive structure than the one ruled out here. The clearest existing model is the CSP dichotomy program: frontier theorems in the present setting will likely require algebraic or comparably global invariants that survive quotient semantics without collapsing to the direct closure-invariant structural regime ruled out here.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Codex, Claude Code, Augment, Kilo, and OpenCode in order to refine prose, clean up notation, edit the LaTeX source, and test alternative phrasings and structural arrangements. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Generative AI tools were used throughout the manuscript, across all sections and across all stages from initial drafting to final revision. The tools were used for boilerplate generation, prose and notation refinement, LaTeX/structure cleanup, translation of informal proof ideas into candidate formal artifacts (Lean/LaTeX), and repeated adversarial reviewer-style critique passes to identify blind spots and clarity gaps.

The author retained full intellectual and editorial control, including problem selection, theorem statements, assumptions, novelty framing, acceptance criteria, and final inclusion/exclusion decisions. No technical claim was accepted solely from AI output. Formal claims reported as machine-verified were admitted only after Lean verification (no `sorry` in cited modules) and direct author review. The author is solely responsible for all statements, citations, and conclusions.

Acknowledgments

The author thanks Tobias Fritz for identifying the optimizer quotient as the coimage of Opt in **Set** in earlier work [25].

References

[1] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM*, 61(1):3:1–3:19, 2014.

[2] Christian Bessiere, Clement Carbonnel, Emmanuel Hebrard, George Katsirelos, and Toby Walsh. Detecting and exploiting subproblem tractability. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 468–474, 2013.

[3] David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953.

[4] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330. IEEE, 2017.

[5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[6] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[7] Ronald A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A*, 222:309–368, 1922.

[8] Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Discovering archipelagos of tractability for constraint satisfaction and counting. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1670–1681. SIAM, 2016.

[9] Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

[10] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[11] Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

[12] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[13] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4765–4774, 2017.

[14] John Myhill and John C. Shepherdson. Effective operations on partial recursive functions. *Zeitschrift f"ur mathematische Logik und Grundlagen der Mathematik*, 1(5):310–317, 1955.

[15] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[16] Zdzisław Pawlak. Rough sets. *International Journal of Parallel Programming*, 11(5):341–356, 1982.

[17] Balaraman Ravindran. *An Algebraic Approach to Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2004.

[18] Henry Gordon Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366, 1953.

[19] Leonard J. Savage. *The Foundations of Statistics*. John Wiley & Sons, 1954.

[20] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226. ACM, 1978.

[21] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[22] Claude E. Shannon. Zero-error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956.

[23] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 7:142–163, 1959. Part 4.

[24] Norman Shapiro. Degrees of computability. *Transactions of the American Mathematical Society*, 82(2):281–299, 1956.

[25] Tristan Simas. The optimizer quotient and the certification trilemma. *arXiv:2603.14689*, 2026. https://arxiv.org/abs/2603.14689.

[26] Roman Slowinski and Daniel Vanderpooten. Decision under uncertainty: A rough set approach. *European Journal of Operational Research*, 80(2):277–289, 1995.

[27] Ryan Williams, Carla P. Gomes, and Bart Selman. Backdoors to typical case complexity. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1173–1178. Morgan Kaufmann Publishers Inc., 2003.

[28] Dmitriy Zhuk. A proof of csp dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342. IEEE, 2017.

Supplementary Material for "Exact Structural Abstraction and Tractability Limits"

Tristan Simas

April 14, 2026

The archived artifact for this paper is available at `https://doi.org/10.5281/zenodo.19457896`.

1 Full Lean Handle Ledger

This supplement provides the complete Lean handle ledger cited by the manuscript. It includes every handle identifier, declaration name, and source module path.

ID	Lean Handle / Source	ID	Lean Handle / Source
FR1	currentFamilies_partition_by_role Paper4dFrontier/Classification.lean	FR2	currentFamilies_partition_counts Paper4dFrontier/Classification.lean
FR3	current_families_have_finite_basis Paper4dFrontier/Classification.lean	FR4	current_positive_interfaces_factor_through_role Paper4dFrontier/Classification.lean
FR5	lifted_families_reduce_to_core Paper4dFrontier/Classification.lean	FR6	degenerateFamilies_complete Paper4dFrontier/Classification.lean
FR7	exists_decisionProblem_realizing_labeling Paper4dFrontier/Realizability.lean	FR8	realizingProblem_decisionEquiv_iff Paper4dFrontier/Realizability.lean
FR13	isOptimal_positiveAffineTransform_iff Paper4dFrontier/FamilyAxioms.lean	FR14	opt_eq_positiveAffineTransform Paper4dFrontier/FamilyAxioms.lean
FR15	isSufficient_positiveAffineTransform_iff Paper4dFrontier/FamilyAxioms.lean	FR16	isRelevant_positiveAffineTransform_iff Paper4dFrontier/FamilyAxioms.lean
FR17	decisionEquiv_relabelActions_iff Paper4dFrontier/FamilyAxioms.lean	FR18	decisionEquiv_relabelStates_iff Paper4dFrontier/FamilyAxioms.lean
FR19	isSufficient_relabelActions_iff Paper4dFrontier/FamilyAxioms.lean	FR20	isSufficient_relabelStates_iff Paper4dFrontier/FamilyAxioms.lean
FR21	allProblems_relabelInvariant Paper4dFrontier/FamilyAxioms.lean	FR22	allProblems_kernelUniversal Paper4dFrontier/FamilyAxioms.lean
FR23	relabelInvariant_not_enough Paper4dFrontier/FamilyAxioms.lean	FR24	optRangeCard_realizingIdentity Paper4dFrontier/FamilyAxioms.lean
FR25	currentFamilies_explicit Paper4dFrontier/Classification.lean	FR26	coreFamilies_explicit Paper4dFrontier/Classification.lean
FR27	liftedFamilies_explicit Paper4dFrontier/Classification.lean	FR28	degenerateFamilies_explicit Paper4dFrontier/Classification.lean
FR29	primitiveMechanisms_explicit Paper4dFrontier/Classification.lean	FR30	isSufficient_iff_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR31	isRelevant_iff_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean	FR32	isIrrelevant_iff_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR33	isMinimalSufficient_iff_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean	FR34	sufficientSets_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR35	relevantSet_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean	FR37	quotientCard_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR38	quotientCard_realizingIdentity Paper4dFrontier/FamilyAxioms.lean	FR39	realizingProblemQuotientEquivLabelRange_apply_quotientMap Paper4dFrontier/Realizability.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR40	realizingProblem_quotientCard_eq_labelRangeCard Paper4dFrontier/Realizability.lean	FR41	certificationStatistic_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR42	sufficientSetCount_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean	FR43	minimalSufficientSetCount_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean
FR44	relevantCoordCount_eq_of_decisionEquiv_iff Paper4dFrontier/FamilyAxioms.lean	FR45	setoidRealizingProblem_decisionEquiv_iff Paper4dFrontier/Realizability.lean
FR46	setoidRealizingProblemQuotientEquivSetoidQuotient_apply_quotientMap Paper4dFrontier/Realizability.lean	FR47	exists_decisionProblem_realizing_setoid Paper4dFrontier/Realizability.lean
FR48	isSufficient_iff_agreementRel_le_decisionEquiv Paper4dFrontier/FamilyAxioms.lean	FR49	isIrrelevant_iff_sufficient_erase Paper4dFrontier/FamilyAxioms.lean
FR50	isRelevant_iff_not_sufficient_erase Paper4dFrontier/FamilyAxioms.lean	FR51	quotientEquiv_of_decisionEquiv_iff_apply_quotientMap Paper4dFrontier/FamilyAxioms.lean
FR52	empty_sufficient_of_constant_opt Paper4dFrontier/ClosureLaws.lean	FR53	irrelevant_of_constant_opt Paper4dFrontier/ClosureLaws.lean
FR54	decisionEquiv_duplicateAction_iff Paper4dFrontier/ClosureLaws.lean	FR55	isSufficient_duplicateAction_iff Paper4dFrontier/ClosureLaws.lean
FR56	isRelevant_duplicateAction_iff Paper4dFrontier/ClosureLaws.lean	FR57	decisionEquiv_duplicateState_iff Paper4dFrontier/ClosureLaws.lean
FR58	isSufficient_duplicateState_iff Paper4dFrontier/ClosureLaws.lean	FR59	isRelevant_duplicateState_iff Paper4dFrontier/ClosureLaws.lean
FR60	duplicateStateQuotientEquivOriginal_apply_quotientMap Paper4dFrontier/ClosureLaws.lean	FR66	booleanCube_card Paper4dFrontier/EnumerationBounds.lean
FR67	exists_booleanCube_larger_than_monomial Paper4dFrontier/EnumerationBounds.lean	FR68	decisionEquiv_addDuplicateState_iff Paper4dFrontier/ClosureLaws.lean
FR69	isSufficient_addDuplicateState_iff Paper4dFrontier/ClosureLaws.lean	FR70	isRelevant_addDuplicateState_iff Paper4dFrontier/ClosureLaws.lean
FR71	addDuplicateStateQuotientEquivOriginal_apply_quotientMap Paper4dFrontier/ClosureLaws.lean	FR72	some_mem_opt_addDuplicateAction_iff Paper4dFrontier/ClosureLaws.lean
FR73	none_mem_opt_addDuplicateAction_iff Paper4dFrontier/ClosureLaws.lean	FR83	isSufficient_noiseExtended_iff Paper4dFrontier/DimensionalNoiseExtension.lean
FR84	isRelevant_noiseExtended_iff Paper4dFrontier/DimensionalNoiseExtension.lean	FR85	lastCoord_irrelevant_noiseExtended Paper4dFrontier/DimensionalNoiseExtension.lean
FR100	parentTree_realTreewidth_one Paper4dFrontier/ParentTreewidth.lean	FR101	low_rank_only_witness Paper4dFrontier/LowRankOnlyWitness.lean
FR105	separable_only_witness Paper4dFrontier/SeparableOnlyWitness.lean	FR106	bounded_actions_only_witness Paper4dFrontier/BoundedActionsOnlyWitness2.lean
FR108	all_independent_core_mechanisms_nondegenerate Paper4dFrontier/Block2Progress.lean	FR110	parentTreeStructured_implies_treeStructured Paper4dFrontier/ParentTreewidth.lean
FR111	treeStructured_and_unique_parent_iff_parentTreeStructured Paper4dFrontier/ParentTreewidth.lean	FR112	bounded_treewidth_only_witness Paper4dFrontier/BoundedTreewidthOnlyWitness.lean
FR113	symmetry_only_witness Paper4dFrontier/SymmetryOnlyWitness.lean	FR114	weak_tree_structured_not_width_one Paper4dFrontier/LegacyTreeGap.lean
FR115	tree_structure_hardness_bridge Paper4dFrontier/TreeStructureHardnessBridge.lean	FR118	pairCrossDifference_eq_binaryCrossDifference_of_lt Paper4dFrontier/BinaryPairwiseDichotomy.lean
FR119	pairwise_zero_crossDifference_unaryDecomposition Paper4dFrontier/BinaryPairwiseDichotomy.lean	FR120	binary_pairwise_symmetry_dichotomy Paper4dFrontier/BinaryPairwiseDichotomy.lean
FR121	actionGapCrossDifference_eq_binaryCrossDifference_of_lt Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean	FR122	decisionRelevant_zero_actionGap_implies_unaryReduction Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean
FR123	binary_pairwise_symmetry_decision_relevant_dichotomy Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean	FR124	block6_genuine_interaction_dichotomy_obstruction Paper4dFrontier/Block6Obstruction.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR125	decisionRelevantInteractionGraph_addActionOffset Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean	FR126	action_offset_can_force_constant_optimizer Paper4dFrontier/Block6Obstruction.lean
FR127	block6_action_offset_obstruction Paper4dFrontier/Block6Obstruction.lean	FR128	offsetNormalizedDecisionRelevantInteractionGraph_wellDefined Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean
FR129	block6_offset_normalized_obstruction Paper4dFrontier/Block6Obstruction.lean	FR130	neverOptimalGhost_decisionRelevantGraph_eq_top Paper4dFrontier/Block6Obstruction.lean
FR131	neverOptimalGhost_supportedGraph_eq_bot Paper4dFrontier/Block6Obstruction.lean	FR132	support_filtering_removes_known_block6_obstructions Paper4dFrontier/Block6Obstruction.lean
FR133	block6_ghost_action_obstruction Paper4dFrontier/Block6Obstruction.lean	FR134	marginMasking_supportedGraph_eq_top Paper4dFrontier/Block6Obstruction.lean
FR135	marginMasking_zero_sufficient Paper4dFrontier/Block6Obstruction.lean	FR136	block6_optimizer_supported_obstruction Paper4dFrontier/Block6Obstruction.lean
FR137	MarginBounded Paper4dFrontier/DecisionRelevantPairwiseDichotomy.lean	FR138	dominantPair_marginBounded Paper4dFrontier/Block6Obstruction.lean
FR139	dominantPair_supportedGraph_eq_top Paper4dFrontier/Block6Obstruction.lean	FR140	block6_margin_bounded_obstruction Paper4dFrontier/Block6Obstruction.lean
FR145	not_collapseLandscapeInfinity_of_oracle_predicate Paper4dFrontier/MetaCharacterization.lean	FR176	DecisionQuotient.DecisionProblem.quotient_is_coarsest paper4/DecisionQuotient/Quotient.lean
FR177	DecisionQuotient.DecisionProblem.quotient_has_unique_factorization paper4/DecisionQuotient/Quotient.lean	FR178	DecisionQuotient.DecisionProblem.srank_eq_relevant_card paper4/DecisionQuotient/Tractability/StructuralRank.lean
FR179	DecisionQuotient.quotientEntropy_le_srank_binary paper4/DecisionQuotient/Information.lean	FR180	DecisionQuotient.Statistics.fisherMatrix_rank_eq_srank paper4/DecisionQuotient/Statistics/FisherInformation.lean
FR181	DecisionQuotient.Information.compression_below_srank_fails paper4/DecisionQuotient/Information/RDSrank.lean	FR182	DecisionQuotient.Information.srank_bits_sufficient paper4/DecisionQuotient/Information/RDSrank.lean
FR183	DecisionQuotient.Physics.single_future_zero_cost paper4/DecisionQuotient/Physics/WassersteinIntegrity.lean	FR184	DecisionQuotient.Physics.transportCost_pos_of_offDiag paper4/DecisionQuotient/Physics/WassersteinIntegrity.lean
FR185	no_closureInvariant_predicate_of_orbit_gap Paper4dFrontier/ObstructionPredicateCandidates.lean	FR186	no_admissibleNormalizationPredicate_decides_dominantPair Paper4dFrontier/ObstructionPredicateCandidates.lean
FR187	no_admissibleNormalizationPredicate_decides_marginBounded Paper4dFrontier/ObstructionPredicateCandidates.lean	FR188	no_admissibleNormalizationPredicate_decides_ghostActionTwoPairCrossOne Paper4dFrontier/ObstructionPredicateCandidates.lean
FR189	no_admissibleNormalizationPredicate_decides_offsetActionZeroPairCrossOne Paper4dFrontier/ObstructionPredicateCandidates.lean	FR190	admissibleCollapseLandscapeInfinity_full_paper Paper4dFrontier/ObstructionPredicateCandidates.lean
FR191	ClosureSoundPackage Paper4dFrontier/ObstructionPredicateCandidates.lean	FR192	ReasonableGuardrailPackage Paper4dFrontier/ObstructionPredicateCandidates.lean
FR193	no_closureSoundPackage_predicate_decides_four_obstruction_families Paper4dFrontier/ObstructionPredicateCandidates.lean	FR194	no_reasonableGuardrailPackage_predicate_decides_four_obstruction_families Paper4dFrontier/ObstructionPredicateCandidates.lean
FR197	classifier_agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/AdmissibleCharacterization.lean	FR198	no_correctOnDomain_classifier_of_orbit_gap Paper4dFrontier/AdmissibleCharacterization.lean
FR199	correct_classifier_inherits_closureLawInvariant Paper4dFrontier/AdmissibleCharacterization.lean	FR200	admissibleNormalizationPredicate_has_explicit_inhabitants Paper4dFrontier/AdmissibleCharacterization.lean
FR201	closureLawInvariant_of_iff_of_closureEquivalent Paper4dFrontier/AdmissibleCharacterization.lean	FR202	exists_orbit_gap_of_not_closureLawInvariant Paper4dFrontier/AdmissibleCharacterization.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR203	closureLawInvariant_iff_no_orbit_gap Paper4dFrontier/AdmissibleCharacterization.lean	FR204	no_exact_closureLawInvariant_classifier_iff_ exists_orbit_gap Paper4dFrontier/AdmissibleCharacterization.lean
FR205	distinctActionCount Paper4dFrontier/DistinctActionProfiles.lean	FR206	actionCount_profileCompressedSlice Paper4dFrontier/DistinctActionProfiles.lean
FR207	decisionEquiv_profileCompressedSlice_iff Paper4dFrontier/DistinctActionProfiles.lean	FR208	profileCompressedSlice_preserves_ exactCertification Paper4dFrontier/DistinctActionProfiles.lean
FR209	profileCompressedSlice_bounded_actions Paper4dFrontier/DistinctActionProfiles.lean	FR210	OrbitGapOn Paper4dFrontier/AdmissibleCharacterization.lean
FR211	no_orbitGapOn_of_exact_classifiable_by_ closureLawInvariant_onDomain Paper4dFrontier/AdmissibleCharacterization.lean	FR212	exact_classifiable_by_closureLawInvariant_ onDomain_iff_no_orbitGapOn Paper4dFrontier/AdmissibleCharacterization.lean
FR213	no_exact_closureLawInvariant_classifier_ onDomain_iff_orbitGapOn Paper4dFrontier/AdmissibleCharacterization.lean	FR214	boundedPatternScheme_holds_largeActionCount_ iff Paper4dFrontier/AdmissibleCharacterization.lean
FR215	boundedPatternDefinable_eventually_constant_ in_actionCount Paper4dFrontier/AdmissibleCharacterization.lean	FR216	boundedPatternDefinable_largeActionCount_ agrees Paper4dFrontier/AdmissibleCharacterization.lean
FR217	payloadSufficient_iff_realizingProblem_ isSufficient Paper4dFrontier/Realizability.lean	FR218	payloadRelevant_iff_realizingProblem_ isRelevant Paper4dFrontier/Realizability.lean
FR219	payloadIrrelevant_iff_realizingProblem_ isIrrelevant Paper4dFrontier/Realizability.lean	FR220	payloadMinimalSufficient_iff_realizingProblem_ isMinimalSufficient Paper4dFrontier/Realizability.lean
FR221	payloadSufficientSets_eq_realizingProblem_ sufficientSets Paper4dFrontier/Realizability.lean	FR222	payloadRelevantSet_eq_realizingProblem_ relevantSet Paper4dFrontier/Realizability.lean
FR223	feasiblePayloadSufficient_iff_ lawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean	FR224	feasiblePayloadRelevant_iff_ lawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean
FR225	feasiblePayloadIrrelevant_iff_ lawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean	FR226	setValuedPayloadSufficient_iff_ totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean
FR227	setValuedPayloadRelevant_iff_ totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean	FR228	setValuedPayloadIrrelevant_iff_ totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean
FR229	outputSemanticsSufficient_iff_ totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean	FR230	outputSemanticsRelevant_iff_ totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean
FR231	outputSemanticsIrrelevant_iff_ totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean	FR232	approximationSemanticsSufficient_iff_ totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean
FR233	approximationSemanticsRelevant_iff_ totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean	FR234	approximationSemanticsIrrelevant_iff_ totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean
FR235	outputSemanticsSufficientSets_eq_of_ outputSemanticsEquivalent Paper4dFrontier/Realizability.lean	FR236	outputSemanticsRelevantSet_eq_of_ outputSemanticsEquivalent Paper4dFrontier/Realizability.lean
FR237	quotientMap_realizes_setoid_asOutputSemantics Paper4dFrontier/Realizability.lean	FR240	exists_outputSemantics_realizing_setoid Paper4dFrontier/Realizability.lean
FR241	outputSemanticsSetoid Paper4dFrontier/Realizability.lean	FR242	SemanticallyExtensionalMap Paper4dFrontier/Realizability.lean
FR243	semanticallyExtensionalMap_factors_through_ outputSemanticsQuotient Paper4dFrontier/Realizability.lean	FR244	semanticallyExtensionalClaim_factors_through_ outputSemanticsQuotient Paper4dFrontier/Realizability.lean
FR245	optimizerComputation_polytime_ closureLawInvariant Paper4dFrontier/ComputeCostApplications.lean	FR246	optimizerSetPayload_polytime_ closureLawInvariant Paper4dFrontier/ComputeCostApplications.lean
FR247	optimizerSetSearch_polytime_ closureLawInvariant Paper4dFrontier/ComputeCostApplications.lean	FR248	optimizerComputation_polytime_classifier_ agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostApplications.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR249	optimizerComputation_polytime_classifier_agrees_on_closureEquivalent Paper4dFrontier/ComputeCostApplications.lean	FR250	no_correct_optimizerComputation_polytime_classifier_decides_dominantPair Paper4dFrontier/ComputeCostApplications.lean
FR251	no_admissibleNormalizationPredicate_optimizerComputation_polytime_and_dominantPair Paper4dFrontier/ComputeCostApplications.lean	FR252	CorrectnessForcesOrbitAgreementOnDomain Paper4dFrontier/AdmissibleCharacterization.lean
FR253	no_orbitGapOn_of_correct_classifier_onDomain_of_forcedOrbitAgreement Paper4dFrontier/AdmissibleCharacterization.lean	FR254	correct_classifier_onDomain_iff_no_orbitGapOn_of_forcedOrbitAgreement Paper4dFrontier/AdmissibleCharacterization.lean
FR255	no_correct_classifier_onDomain_iff_orbitGapOn_of_forcedOrbitAgreement Paper4dFrontier/AdmissibleCharacterization.lean	FR256	optimizerComputation_polytime_correct_classifier_onDomain_iff_no_orbitGapOn Paper4dFrontier/ComputeCostApplications.lean
FR259	optimizerSetPayload_polytime_classifier_agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostApplications.lean	FR260	optimizerSetPayload_polytime_classifier_agrees_on_closureEquivalent Paper4dFrontier/ComputeCostApplications.lean
FR261	optimizerSetSearch_polytime_classifier_agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostApplications.lean	FR262	optimizerSetSearch_polytime_classifier_agrees_on_closureEquivalent Paper4dFrontier/ComputeCostApplications.lean
FR263	relevant_of_uniformApprox_of_strict_gap_witness Paper4dFrontier/ApproximateAdmissibility.lean	FR264	relevance_can_flip_under_arbitrarily_small_uniform_perturbation Paper4dFrontier/ApproximateAdmissibility.lean
FR266	not_sufficient_of_uniformApprox_of_strict_gap_witness Paper4dFrontier/ApproximateAdmissibility.lean	FR267	sufficiency_can_flip_under_arbitrarily_small_uniform_perturbation Paper4dFrontier/ApproximateAdmissibility.lean
FR268	pacGuaranteeSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean	FR269	pacGuaranteeSemanticsRelevant_iff_totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean
FR270	pacGuaranteeSemanticsIrrelevant_iff_totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean	FR271	regretGuaranteeSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean
FR272	regretGuaranteeSemanticsRelevant_iff_totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean	FR273	regretGuaranteeSemanticsIrrelevant_iff_totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean
FR274	statisticalRiskSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean	FR275	statisticalRiskSemanticsRelevant_iff_totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean
FR276	statisticalRiskSemanticsIrrelevant_iff_totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean	FR277	anytimeGuaranteeSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean
FR278	anytimeGuaranteeSemanticsRelevant_iff_totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean	FR279	anytimeGuaranteeSemanticsIrrelevant_iff_totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean
FR280	finiteHorizonGuaranteeSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean	FR281	finiteHorizonGuaranteeSemanticsRelevant_iff_totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean
FR282	finiteHorizonGuaranteeSemanticsIrrelevant_iff_totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean	FR283	statisticalGuaranteeSemanticsSufficientSets_eq_of_outputSemanticsEquivalent Paper4dFrontier/Realizability.lean
FR284	statisticalGuaranteeSemanticsRelevantSet_eq_of_outputSemanticsEquivalent Paper4dFrontier/Realizability.lean	FR285	statisticalGuarantee_classifier_agrees_on_closureEquivalent_of_correctOnDomain_of_transfer Paper4dFrontier/StatisticalSemanticsApplications.lean
FR286	no_correctOnDomain_statisticalGuarantee_classifier_of_orbit_gap_of_transfer Paper4dFrontier/StatisticalSemanticsApplications.lean	FR287	statisticalGuarantee_correct_classifier_onDomain_iff_no_orbitGapOn_of_transfer Paper4dFrontier/StatisticalSemanticsApplications.lean
FR288	opt_eq_of_uniformApprox_of_uniformStrictGapCover Paper4dFrontier/ApproximateAdmissibility.lean	FR289	sufficientSets_eq_of_uniformApprox_of_uniformStrictGapCover Paper4dFrontier/ApproximateAdmissibility.lean
FR290	relevantSet_eq_of_uniformApprox_of_uniformStrictGapCover Paper4dFrontier/ApproximateAdmissibility.lean	FR291	randomizedGuaranteeSemanticsSufficient_iff_totalizedLawDecisionProblem_isSufficient Paper4dFrontier/Realizability.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR292	randomizedGuaranteeSemanticsRelevant_iff_ totalizedLawDecisionProblem_isRelevant Paper4dFrontier/Realizability.lean	FR293	randomizedGuaranteeSemanticsIrrelevant_iff_ totalizedLawDecisionProblem_isIrrelevant Paper4dFrontier/Realizability.lean
FR294	randomizedGuarantee_classifier_agrees_on_ closureEquivalent_of_correctOnDomain_of_ transfer Paper4dFrontier/StatisticalSemanticsApplications.lean	FR295	no_correctOnDomain_randomizedGuarantee_ classifier_of_orbit_gap_of_transfer Paper4dFrontier/StatisticalSemanticsApplications.lean
FR296	randomizedGuarantee_correct_classifier_ onDomain_iff_no_orbitGapOn_of_transfer Paper4dFrontier/StatisticalSemanticsApplications.lean	FR297	transferredSemantics_classifier_agrees_on_ closureEquivalent_of_correctOnDomain Paper4dFrontier/StatisticalSemanticsApplications.lean
FR298	no_correctOnDomain_transferredSemantics_ classifier_of_orbit_gap Paper4dFrontier/StatisticalSemanticsApplications.lean	FR299	transferredSemantics_correct_classifier_ onDomain_iff_no_orbitGapOn Paper4dFrontier/StatisticalSemanticsApplications.lean
FR300	decisionEquiv_iff_of_uniformApprox_of_ uniformStrictGapCover Paper4dFrontier/ApproximateAdmissibility.lean	FR301	isMinimalSufficient_iff_of_uniformApprox_of_ uniformStrictGapCover Paper4dFrontier/ApproximateAdmissibility.lean
FR302	outputSemanticsExactRelevanceProfile_eq_of_ outputSemanticsEquivalent Paper4dFrontier/Realizability.lean	FR303	outputSemanticsExactRelevanceProfile_eq_ totalizedLawDecisionProblem Paper4dFrontier/Realizability.lean
FR304	agreeOn_univ_iff_eq_of_ finiteIndicatorCoordinateSpace Paper4dFrontier/Realizability.lean	FR305	finite_outputSemantics_ allCoordinatesSufficient Paper4dFrontier/Realizability.lean
FR306	finite_outputSemantics_realized_by_ exactCertification Paper4dFrontier/Realizability.lean	FR307	agreeOn_univ_iff_eq_of_ singletonIdentityCoordinateSpace Paper4dFrontier/Realizability.lean
FR308	outputSemantics_admits_coordinatePresentation Paper4dFrontier/Realizability.lean	FR311	encodable_stateSpace_admits_ countableBooleanPresentation Paper4dFrontier/Realizability.lean
FR312	encodable_discreteMeasurable_stateSpace_ admits_measurable_countableBooleanPresentation Paper4dFrontier/Realizability.lean	FR313	encodable_discreteTopological_stateSpace_ admits_continuous_countableBooleanPresentation Paper4dFrontier/Realizability.lean
FR314	agreeOn_univ_iff_eq_of_ finiteBinaryCoordinateSpace Paper4dFrontier/Realizability.lean	FR315	finite_exactSpecification_admits_ lowDimBooleanPresentation Paper4dFrontier/Realizability.lean
FR316	IdentityOutputClosureSpec.classifier_agrees_ on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR317	IdentityOutputClosureSpec.no_correctOnDomain_ classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean
FR318	hypothesisOutput_classifier_agrees_on_ closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR319	estimatorOutput_classifier_agrees_on_ closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean
FR320	policyOutput_classifier_agrees_on_ closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR321	randomizedProcedure_classifier_agrees_on_ closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean
FR322	no_correctOnDomain_hypothesisOutput_ classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean	FR323	no_correctOnDomain_estimatorOutput_classifier_ of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean
FR324	no_correctOnDomain_policyOutput_classifier_of_ orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean	FR325	no_correctOnDomain_randomizedProcedure_ classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean
FR326	TransportedOutputClosureSpec.classifier_ agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR327	TransportedOutputClosureSpec.no_ correctOnDomain_classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean
FR328	IdentityOutputClosureSpec.correctOnDomain_iff_ correctOnDomain_transport Paper4dFrontier/ComputeCostExternalOutputs.lean	FR329	representationRelativeHypothesis_classifier_ agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean
FR330	representationRelativeEstimator_classifier_ agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR331	representationRelativePolicy_classifier_ agrees_on_closureEquivalent_of_correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean
FR332	representationRelativeRandomizedProcedure_ classifier_agrees_on_closureEquivalent_of_ correctOnDomain Paper4dFrontier/ComputeCostExternalOutputs.lean	FR333	no_correctOnDomain_representationRelativeHypothesis_ classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean

(continued...)

ID	Lean Handle / Source	ID	Lean Handle / Source
FR334	no_correctOnDomain_representationRelativeEstimator_classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean	FR335	no_correctOnDomain_representationRelativePolicy_classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean
FR336	no_correctOnDomain_representationRelativeRandomizedProcedure_classifier_of_orbit_gap Paper4dFrontier/ComputeCostExternalOutputs.lean	FR340	booleanPayloadTransfer Paper4dFrontier/Realizability.lean
FR341	predicateTransfer Paper4dFrontier/Realizability.lean	FR342	sufficiency_is_relation_refinement Paper4dFrontier/Realizability.lean
FR343	relevance_is_erased_failure_of_refinement Paper4dFrontier/Realizability.lean	FR345	exactSemanticsQuotient_universal_characterization Paper4dFrontier/Realizability.lean
FR346	exactnessMeansExactAgreementWithValidity Paper4dFrontier/Realizability.lean		

2 Claim-to-Lean Handle Mapping

This section maps each paper claim to its corresponding Lean formalization.

Paper claim	Lean handle
Corollary 6.44: Exact Approximation Semantics Transfer	FR234, FR233, FR232
Corollary 6.26: Boolean Payload Transfer	FR340
Corollary 6.6: Constant-Optimizer Collapse Forces Trivial Certification	FR52, FR53
Corollary 4.14: Domain Restriction Helps Only by Removing Orbit Gaps	FR252, FR254, FR255, FR253, FR256
Corollary 6.50: Every State Equivalence Relation Is Realizable as Exact Output Semantics	FR240, FR237
Corollary 6.20: Compute-Cost Version: External Output Objects	FR316, FR328, FR317, FR326, FR327, FR319, FR318, FR323, FR322, FR324, FR325, FR320, FR321
Corollary 2.3: Finite Basis for the Current Positive Landscape	FR3, FR29
Corollary 4.11: No Correct Tractability Classifier	FR197, FR199, FR193, FR198
Corollary 4.13: Orbit-Gap Completeness on Closure-Closed Domains	FR210, FR212, FR213, FR211
Corollary 6.19: Compute-Cost Version: Canonical Payload and Search Tasks	FR260, FR259, FR262, FR261
Corollary 6.27: Predicate Transfer	FR341
Corollary 6.42: Randomized-Output Guarantee Semantics Inherit the Closure-Orbit Consequences	FR295, FR293, FR292, FR291, FR294, FR296
Corollary 6.41: Randomized-Output Guarantee Semantics Transfer	FR293, FR292, FR291
Corollary 5.4: Realizability Alone Cannot Be the Frontier	FR7, FR8
Corollary 4.10: Reasonable Guardrail Packages	FR194
Corollary 6.30: Arbitrary Exact Output Semantics Transfer	FR231, FR230, FR229
Corollary 6.3: Relevance Is Failure of Erased Sufficiency	FR49, FR50
Corollary 6.33: Relevance Is Erased Failure of Refinement	FR343
Corollary 6.28: Nonempty Set-Valued Payload Transfer	FR225, FR224, FR223
Corollary 6.46: Arbitrarily Small Uniform Perturbations Can Flip Relevance	FR264
Corollary 6.47: Arbitrarily Small Uniform Perturbations Can Flip Sufficiency	FR267

Paper claim	Lean handle
Corollary 6.43: Statistical Guarantee Semantics Inherit the Closure-Orbit Consequences	FR279, FR278, FR277, FR282, FR281, FR280, FR295, FR286, FR298, FR288, FR270, FR269, FR268, FR293, FR292, FR291, FR294, FR296, FR273, FR272, FR271, FR290, FR284, FR283, FR285, FR287, FR276, FR275, FR274, FR289, FR297, FR299
Corollary 6.40: Statistical Guarantee Semantics Transfer	FR279, FR278, FR277, FR282, FR281, FR280, FR270, FR269, FR268, FR273, FR272, FR271, FR276, FR275, FR274
Corollary 6.4: Certification Statistics Factor Through the Quotient Relation	FR41, FR43, FR44, FR42
Corollary 6.2: Sufficiency Is Relation Refinement	FR48
Corollary 6.29: Arbitrary Set-Valued Payload Transfer via Failure Tokens	FR228, FR227, FR226
Corollary 6.21: Compute-Cost Version: Representation-Relative Output Objects	FR328, FR326, FR327, FR334, FR333, FR335, FR336, FR330, FR329, FR331, FR332
Proposition 6.22: The Admissibility Layer Is Nonempty	FR200
Proposition 6.8: Positive Affine Utility Reparameterizations Are Invisible	FR13, FR16, FR15, FR14
Proposition 6.45: Approximate Relevance and Sufficiency Claims Need Explicit Stability Control	FR266, FR263
Proposition 4.1: Binary Pairwise Symmetry Dichotomy	FR120, FR118, FR119
Proposition 6.24: Bounded-Pattern Predicates Stabilize Above a Finite Action Bound	FR215, FR216, FR214
Proposition 6.35: Canonical Exact Relevance Profile	FR302, FR303
Proposition 6.15: Closure Operations Preserve Exact Certification	FR57, FR17, FR18, FR60, FR56, FR59, FR84, FR16, FR55, FR58, FR83, FR15, FR19, FR20, FR85
Proposition 6.38: Countable Exact Specifications Admit Countable Boolean Presentations	FR312, FR313, FR311
Proposition 4.2: Decision-Relevant Binary Pairwise Dichotomy	FR121, FR123, FR124, FR122
Proposition 3.2: Degenerate Positive Cases	FR1, FR6
Proposition 6.1: Exact Certification Depends Only on the Decision Quotient Relation	FR32, FR33, FR31, FR30, FR37, FR51, FR35, FR34
Proposition 6.23: Bounded Distinct Action Profiles Compress to Bounded Actions	FR206, FR207, FR205, FR209, FR208
Proposition 6.9: Duplicate Actions and Duplicate States Preserve Certification	FR71, FR68, FR54, FR70, FR56, FR69, FR55, FR73, FR72
Proposition 6.7: Explicit Enumeration Is Parameter-Dependent, Not Structural	FR66, FR67
Proposition 6.36: Every Exact Specification Admits a Coordinate Presentation	FR307, FR308
Proposition 6.31: Exactness Means Exact Agreement With Validity	FR346
Proposition 2.4: Explicit Primitive Basis	FR29
Proposition 2.2: Explicit Inventory	FR26, FR25, FR3, FR28, FR27
Proposition 6.37: Finite Exact Specifications Admit Coordinate Presentations	FR304, FR305, FR306
Proposition 6.39: Finite Exact Specifications Admit Low-Dimensional Boolean Presentations	FR314, FR315

Paper claim	Lean handle
Proposition 2.5: All Independent Core Mechanisms Are Nondegenerate	FR108, FR106, FR112, FR101, FR105, FR113
Proposition 6.48: Global Approximation Stability Under Uniform Strict Gaps	FR300, FR301, FR295, FR298, FR288, FR293, FR292, FR291, FR294, FR296, FR290, FR289, FR297, FR299
Proposition 2.9: Current Positive Interfaces Factor Through Role Classification	FR4
Proposition 6.12: Invariance Alone Is Too Weak	FR22, FR21, FR23
Proposition 2.6: Weak Tree Ordering Does Not Imply Width One	FR115, FR114
Proposition 3.1: Lifted Cases Introduce No New Primitive	FR5
Proposition 6.14: Unrestricted Predicates Trivialize Exact Characterization	FR145
Proposition 4.12: Orbit-Gap Completeness for Exact Classification	FR203, FR201, FR202, FR204
Proposition 4.3: Orbit-Gap Template	FR185
Proposition 2.8: Parent-Tree Structure Gives Width-One Decompositions	FR110, FR100, FR111
Proposition 6.25: Deterministic Payload Transfer	FR219, FR220, FR222, FR218, FR221, FR217
Proposition 6.11: Quotient Size Is Unbounded Under Realizability	FR24, FR38
Proposition 5.3: The Realized Quotient Is Exactly the Label Range	FR39, FR40
Proposition 6.10: Relabeling Invariance Is Forced by Exact Certification	FR17, FR18, FR19, FR20
Proposition 6.49: Exact Semantic Claims Depend Only on Admissible-Output Equivalence	FR236, FR235
Proposition 6.51: Semantically Extensional Claims Factor Through the Exact-Semantics Quotient	FR242, FR241, FR244, FR243
Proposition 5.2: Every Equivalence Relation Is Optimizer-Realizable	FR47, FR46, FR45
Proposition 6.13: Independent Summary Frameworks Converge on the Same Structural Core	FR178, FR181, FR182, FR180, FR179
Proposition 6.32: Sufficiency Is Relation Refinement	FR342
Proposition 2.7: Cyclic Dependencies Recover Hardness	FR110, FR100, FR111, FR115
Proposition 6.5: Zero-Distortion Summaries Refine the Optimizer Quotient	FR177, FR176
Theorem 4.8: Admissible Collapse Across the Four Obstruction Families	FR190
Theorem 4.9: Closure-Sound Package No-Go	FR193
Definition 6.17: Admissible Normalization Predicate	FR197, FR199, FR198, FR249, FR248, FR245, FR246, FR247
Theorem 2.1: Explicit Partition of the Current Positive Landscape	FR26, FR25, FR1, FR2, FR28, FR27
Theorem 4.4: Dominant-Pair Admissible No-Go	FR186
Theorem 6.34: Exact Semantics Quotient Universality	FR345
Theorem 4.6: Ghost-Action Admissible No-Go	FR188
Theorem 5.1: Every Labeling Kernel Is Optimizer-Realizable	FR7, FR8
Theorem 4.5: Margin-Masking Admissible No-Go	FR187
Theorem 4.7: Offset Admissible No-Go	FR189
Theorem 6.18: Compute-Cost Version: Optimizer Computation	FR248

Auto summary: mapped 79/79 (full=79, derived=0, unmapped=0).