Verdict Calculus: A Deterministic Judgment System for Semantic Regimes

Adrian Diamond

February 7, 2026

Abstract

We introduce *Verdict Calculus*, a formal judgment system for assigning well-typed verdicts to semantic classifications produced by semantic regime systems. The calculus operates over finite semantic domains and produces deterministic, non-executing verdict artifacts suitable for governance, audit, and policy compilation. Verdict Calculus is positioned as a downstream semantic layer relative to Semantic Regime Theory and systems such as EGON.

Contents

1 Introduction

Semantic systems increasingly produce structured classifications rather than raw predictions. While such classifications are deterministic and interpretable, they often lack a formal judgment layer specifying whether a classification is acceptable, actionable, neutral, or escalatory.

Verdict Calculus addresses this gap by providing a formal, type-theoretic judgment system over semantic regimes. The calculus does not execute actions, infer outcomes, or learn from data. It exists solely to assign *verdicts* to semantic results.

2 Semantic Inputs

Definition 2.1 (Semantic Domain)**.** Let F be a finite set of semantic classes produced by a semantic regime system.

Definition 2.2 (Semantic Observation)**.** A semantic observation is an element $f \in F$.

Verdict Calculus assumes semantic correctness of its inputs. It does not re-evaluate or reinterpret the classification process.

3 Verdict Judgments

Definition 3.1 (Verdict Set)**.** Let V be a finite set of verdicts, such as

$$V = \{\mathsf{Accept}, \mathsf{Observe}, \mathsf{Escalate}, \mathsf{Block}, \mathsf{Neutral}\}.$$

Definition 3.2 (Verdict Judgment)**.** A verdict judgment has the form

$$\Gamma \vdash f \Rightarrow v$$

where $f \in F$, $v \in V$, and Γ is a typing environment.

Remark 3.3. Verdict judgments are declarative. They do not imply execution or enforcement.

4 Typing Environments

Definition 4.1 (Typing Environment)**.** A typing environment Γ is a finite context specifying constraints, capabilities, or governance conditions under which verdicts are evaluated.

Definition 4.2 (Verdict Function)**.** A verdict function is a deterministic map

$$\mathcal{V}_\Gamma : F \to V.$$

Theorem 4.3 (Determinism)**.** *For fixed Γ and $f \in F$, the verdict $\mathcal{V}_\Gamma(f)$ is unique.*

5 Soundness and Totality

Definition 5.1 (Totality)**.** Verdict Calculus is total if every semantic input admits a verdict.

Theorem 5.2 (Totality)**.** *For all $f \in F$, there exists a verdict $v \in V$ such that*

$$\Gamma \vdash f \Rightarrow v.$$

Remark 5.3. Neutral verdicts are explicit elements of V, ensuring that non-action is a first-class outcome.

6 Relationship to Semantic Regime Theory

Verdict Calculus operates strictly downstream of Semantic Regime Theory. While SRT defines how metric or signal-based spaces collapse into semantic regimes, Verdict Calculus assigns judgments to those regimes.

The calculus neither modifies nor refines semantic classification. It interprets semantic outputs within a governance or policy context.

7 What Verdict Calculus Is Not

Verdict Calculus is not:

- a prediction system,
- an optimization framework,
- a control algorithm,
- an execution engine,
- a learning system.

Its sole purpose is deterministic judgment.

8 Future Directions

Future work includes:

- a verdict intermediate representation (IR),
- a reference checker implementation,
- compilation of verdicts into policy artifacts,
- integration with EGON-style semantic engines.

These extensions preserve the non-executing, declarative nature of the calculus.

Conclusion

Verdict Calculus provides a principled judgment layer for semantic systems. By separating semantic classification from judgment, it enables deterministic governance, auditability, and policy compilation without sacrificing interpretability.