

Constraint-Emergence Ontology

A foundational framework proposing that reality, computation, and engineered systems share structural invariants

Introduction

This repository contains a philosophical ontology and its worked applications. The central thesis: reality is fundamentally a self-organising constraint network. Stable patterns — Markov objects — emerge in gaps between constraints, and their boundaries, hierarchies, and dynamics recur across substrates (physics, computation, biology, cognition, engineered systems).

The framework operates at the level of structure, not material:

The invariants of reality live in the structure of admissible transformations, not in the material being transformed.

From this ontology, two practical contributions follow:

1. **Emergent Reasoning** — a formal model of LLM computation as constrained topological traversal on a semantic manifold, explaining how probabilistic systems produce structured inference.
2. **Logical Encapsulation** — a method for programming LLM reasoning by loading constraint specifications (axioms, invariants, evaluation algorithms) rather than detailed instructions. This converts an LLM from a generative peer into a mechanical evaluator.

The Political OS Suite is the primary worked example of Logical Encapsulation: four competing political philosophies expressed as formal constraint specifications, each producing mechanically divergent analyses of the same political phenomena.

Repository Structure

```
constraint_emergence_ontology/  
├─ constraint_emergence_ontology.md    # Core ontology  
├─ concepts.md                        # Concept index, dependencies,  
and status map  
├─ ontology_templates.md              # Logical Encapsulation meta-  
template  
├─ presentations/                     # PDF snapshots (periodically  
updated)  
└─ ── constraint_emergence_ontology.pdf
```

```

├── ontology_templates.pdf
├── README.pdf
└── political_os/                                # Worked example: Political OS
Suite
  ├── README.md                                # Political OS introduction and
reading guide
  ├── classical_liberal_political_os.md
  ├── marxist_political_os.md
  ├── critical_justice_political_os.md
  ├── theocratic_political_os.md
  ├── us_democratic_political_os.md
  ├── political_operating_system.md           # Main paper – start here
  ├── political_os_test_suite.md
  └── presentations/                          # PDF snapshots (periodically
updated)
    ├── classical_liberal_political_os.pdf
    ├── marxist_political_os.pdf
    ├── critical_justice_political_os.pdf
    ├── theocratic_political_os.pdf
    ├── us_democratic_political_os.pdf
    ├── political_operating_system.pdf
    ├── political_os_test_suite.pdf
    └── README.pdf
└── reports/                                  # Real-world invariant analyses
(.md + .pdf)
  ├── 2026-02-16-australia-invariant-analysis.md
  ├── 2026-02-16-uk-invariant-analysis.md
  ├── 2026-02-16-canada-invariant-analysis.md
  ├── 2026-02-16-germany-invariant-analysis.md
  ├── 2026-02-16-united-states-invariant-analysis.md
  └── 2026-02-16-california-invariant-analysis.md

```

Documents

Core Framework

Document	Description
<u>Constraint Emergence Ontology</u>	The core philosophical work (v1.3). Constraint networks, Markov objects, emergent manifolds, observer theory, meaning as structural invariant, information-driven construction pattern. Part VIII-D formalizes the Constraint Functor — the category-theoretic bridge between physical and computational Markov objects.
<u>Concept Index</u>	Canonical concept map for the ontology: definitions, section locations, dependency

Document	Description
	graph, and epistemic status tags (axiom, derived, conjecture, instantiation).
<u>Emergent Reasoning</u>	Formal companion paper (separate repo). LLMs as constraint-manifold traversal systems: attention as soft unification, proto-symbolic attractors, hallucination as probability degeneracy. Published on Zenodo .
<u>Ontology Templates</u>	The Logical Encapsulation meta-template. How to build constraint specifications that program LLM reasoning within defined axioms and procedures. Published on Zenodo .

Political OS Suite

Four political philosophies expressed as formal constraint specifications. Start with **The Political Operating System** — the main paper introducing the Governance Stack model, structural comparison, and key findings. Then load individual OS specifications into an LLM to see them in action.

Document	Nature
<u>The Political Operating System</u>	Entry point — Governance Stack, structural comparison, key findings
<u>Classical Liberal OS</u>	Full governance specification
<u>Marxist OS</u>	Diagnostic with governance gap
<u>Critical Justice OS</u>	Diagnostic program
<u>Theocratic OS</u>	Full governance specification

How to Read This

If you want to understand the ontology

1. Start with **Constraint-Emergence Ontology**. Read Part 0 (structural invariance) and Part I (the ontology itself — sections 1-18). Part II positions against existing thinkers; Part VIII maps to specific domains; Part IX is the research agenda.
2. Use **Concept Index** as the glossary and dependency map while reading; it tracks definitions, section anchors, and concept status.
3. Read **Emergent Reasoning** for the formal treatment of how LLMs instantiate the constraint architecture.

If you want to see the method in action

1. Read [Ontology Templates](#) to understand Logical Encapsulation.
2. Go to the [Political OS Suite](#) — follow its README for how to load and test the constraint specifications.

If you want to understand the Political OS

Start with [The Political Operating System](#) — it frames the entire suite. See the [Political OS README](#) for quick start, test suite instructions, and real-world analysis reports.

Related Work

- [ai_sdgc_method](#) — The AI SDLC methodology providing the software engineering empirical ground referenced in Part VIII
- [emergent_reasoning](#) — Extended analysis, simulations, and peer review of the emergent reasoning paper

Publication

- Emergent Reasoning paper: [Zenodo](#)
- Constraint-Emergence Ontology: [Zenodo](#)
- Programming LLM Reasoning (Ontology Templates): [Zenodo](#)

Author

Dimitar Popov

License

This work is shared for academic and philosophical discussion.