World-Model Calculus for Probabilistic Logic Networks: Evidence, Views, and Σ-Guarded Compiled Inference

Zar (project direction) Oruži (compiled notes)

February 6, 2026

Abstract

This note distills the core ideas developed while formalizing a *World-Model (WM) calculus* for Probabilistic Logic Networks (PLN) inside Lean (the `Mettapedia` repository).

The guiding principle is: *PLN inference is stateful posterior revision plus evidence extraction*, and "fast PLN rules" are *compiled, Σ-guarded rewrites* whose soundness is discharged relative to a declared *world-model class* (Bayesian network, factor graph, joint evidence, etc.).

We separate: (i) **evidence** as a rich algebraic object (often Heyting/quantale-valued), (ii) **views** (probabilities, STVs/WTVs, typicalities) as explicit projections from evidence, and (iii) **tractable sublayers** where inference is exact relative to assumptions (e.g. d-separation). We explain how this architecture supports message passing / activation spreading (MORK-style), how to make approximation risk explicit (ledgers, hypothesis weights, projection operators), and how to integrate **quantifiers** via satisfying-set semantics and Goertzel-style weakness. Finally we outline a proof-calculus agenda: soundness and (fragmented) completeness results relative to chosen semantics (propositional/FO/Henkin-HO, finite vs quasi-Borel, etc.).

Contents

1 Motivation and the "council direction"

PLN aims to support *probabilistic reasoning in open-ended domains*: uncertain facts, imperfect independence, and large-scale inference control (message passing, GNN-like spreading, positive feedback loops). The historical friction is well-known:

- **Local truth-value rules are not globally complete.** Even if you have local evidence for A, B, C and for links $A \Rightarrow B$, $B \Rightarrow C$, you cannot in general compute the *exact* evidence for $A \Rightarrow C$ without carrying joint correlation information.
- **Independence assumptions are the real price.** In practice you need to ask: when is a Markov/d-separation assumption justified, what do you do when it is violated, and how do you quantify drift?
- **Operational factor graphs are not automatically semantics.** A message-passing graph whose factors are "deduction" or "induction" update formulas is useful for search, but it is only semantically correct when those factors are *compiled* from a true world-model factorization (or guarded by explicit assumptions).

The emerging design answer is:

> **Keep the semantic kernel small and honest.** World-model states revise under evidence, and queries extract evidence. Everything else—truth values, probabilities, "fast rules", message passing—are *explicit views and* Σ*-guarded tactics* whose validity is proved *relative to a declared model class.*

This aligns the "council" instincts: Kolmogorov/Cox/Knuth–Skilling give valuation axioms, Kohlas–Shenoy give algebraic computation, Tao wants hypotheses stated cleanly, and Meredith/Stay want the categorical/process structure to stay compositional.

2 The WM kernel: states, revision, and queries

2.1 World-model interface

At the semantic core we posit:

- a type of world-model states $W : \mathsf{State}$ (posterior states, knowledge states, factorized models, etc.),
- a type of queries $q : \mathsf{Query}$ (propositions, links, quantified claims, etc.),
- an evidence type $\mathsf{Evidence}$, and
- operations *revision* and *evidence extraction*.

Conceptually:

$W \oplus \Delta$ is revision of state W by new evidence Δ $\quad \mathsf{evidence}(W, q) \in \mathsf{Evidence}$ is evidence supporting query q in

In Lean this is packaged as a `WorldModel` typeclass with an `EvidenceType` for states. The essential law is **additivity of evidence extraction under revision**:

$$\mathsf{evidence}(W_1 \oplus W_2, q) = \mathsf{evidence}(W_1, q) \oplus \mathsf{evidence}(W_2, q).$$

This is the "semantic spine" that makes revision and querying behave like a calculus.

2.2 Judgments as a proof-calculus skeleton

The WM kernel naturally yields sequent-style judgments:

- $\vdash_{\mathsf{WM}} W$ — *W is a derivable/valid WM state* (a posterior produced from priors and updates),
- $\vdash_q W \Downarrow q \mapsto e$ — *from state W, query q yields evidence e*,
- optionally $\vdash_s W \Downarrow q \mapsto s$ — a scalar *strength view* of the query.

This is already a *proof calculus*, but it is a *stateful* one: proof objects are states and evidence terms, not only formulas. To avoid re-introducing "hidden semantics" through convenience truth-values, we treat STVs/WTVs/probabilities as **views** computed from evidence.

2.3 Query equivalence and Σ-guarded rewrites

We isolate a key metatheoretic concept:

Definition 1 (Query equivalence)**.** *Two queries $q_1, q_2 \in \mathsf{Query}$ are* evidence-equivalent *if*

$$\forall W \in \mathsf{State}.\ \mathsf{evidence}(W, q_1) = \mathsf{evidence}(W, q_2).$$

A "PLN inference rule" is then packaged as a *sound rewrite*:

Definition 2 (Σ-guarded rewrite rule)**.** *A rewrite rule consists of:*

- *a side condition Σ (a proposition or a structured context),*
- *a conclusion query q,*
- *a derived evidence term* $\mathrm{derive}(W)$*, and*
- *a proof that if Σ holds then* $\mathrm{derive}(W) = \mathsf{evidence}(W, q)$ *for all W.*

In practice, many PLN rules rewrite only the *strength view* of a query: they prove an identity about $\mathsf{strength}(\mathsf{evidence}(W, q))$ rather than about full evidence.

3 Evidence vs views: keep approximations explicit

3.1 Evidence as an algebraic object

PLN needs an evidence object that can encode:

- positive and negative support,
- partial truth / paraconsistency,
- monotone aggregation of evidence, and
- (in some sublayers) probabilistic conjugacy (e.g. Beta/Dirichlet counts).

A useful abstraction is that $\mathsf{Evidence}$ forms (at least) a *commutative quantale*: a complete lattice with a monoidal product $\otimes$ distributing over joins. In several PLN developments, $\mathsf{Evidence}$ is also (or instead) treated as a Heyting algebra (so negation is intuitionistic/paraconsistent rather than Boolean).

3.2 Views as projections

A **view** is a function $v : \mathsf{Evidence} \to X$ (often $X = [0,1]$ or $\mathbb{R}^{\infty}_{\geq 0}$), chosen for operational convenience:

$$\text{probability view} \quad p(e) \in [0,1], \qquad \text{STV/WTV view} \quad \mathrm{stv}(e) \in [0,1] \times [0,1], \quad \text{etc.}$$

Views *must be explicit*, because they necessarily forget information. This is not "annoying monadic programming" for its own sake—it is the enforcement mechanism that prevents silent unsoundness.

Remark 1 (The totality gate)**.** *If* $\mathsf{Evidence}$ *has incomparable elements, there is no faithful order-embedding into* $\mathbb{R}$*. So a single real-valued view cannot be globally complete. This is not a bug; it is a statement about what information you are discarding.*

4 A key no-go: why completeness cannot be purely local

A foundational result proved in the Lean development can be phrased as:

Theorem 1 (No local complete link-deduction rule (informal))**.** *There is no function*

$$f : \mathsf{Evidence}^5 \to \mathsf{Evidence}$$

that takes only local *link/proposition evidence (for A, B, C and $A \Rightarrow B$, $B \Rightarrow C$) and returns the* exact *evidence for $A \Rightarrow C$ for* all *world-model states.*

Interpretation: **correlations live in the joint state**. So "fast PLN rules" must be framed as *admissible rewrites under assumptions*: they are not globally complete inference rules in the sequent-calculus sense.

This no-go result is the reason the WM calculus is *state-first*: the state carries whatever correlation structure is needed for truth of queries.

5 Tractable sublayers: Bayesian networks and factor graphs

5.1 World-model classes and Σ side conditions

To scale, we restrict the class of admissible states:

$$\mathcal{C} \subseteq \mathsf{State} \quad \text{(e.g. DAG-factorizing BNs, Markov random fields, etc.).}$$

Inside such a class, we can prove additional theorems (screening-off, d-separation), and compile them into admissible rewrite rules.

We write Σ for explicit side conditions:

- structural hypotheses (a particular DAG/factorization),
- conditional independence facts (from d-separation),
- positivity/nondegeneracy assumptions (to avoid division by 0),
- approximation/error ledger invariants (if using heuristics).

5.2 Variable elimination as the exact query engine

Given a discrete factor graph (or a BN compiled to a factor graph), we can compute unnormalized weights for constraints by variable elimination (VE):

$$Z(\text{constraints}) \;=\; \sum_{\text{assignments consistent with constraints}} \prod_{f} \phi_f.$$

In Lean, a generic VE engine is implemented for factor graphs parameterized by a semiring K, with specializations to $\mathbb{R}^{\infty}_{\geq 0}$ (ENNReal) for probabilities.

Remark 2. *A `CommSemiring` assumption is* stronger *than separate `AddCommMonoid+Mul`, but it is not "making it a ring". Rings add additive inverses; semirings do not. We use semirings because VE is "sum of products" and needs distributivity.*

5.3 Fast PLN rules as compiled tactics

A PLN rule (e.g. deduction) becomes:

> **A Σ-guarded rewrite of a hard query into smaller queries**, together with a proof that the *strength view* of the conclusion equals the rule formula applied to the strength views of the premises.

Crucially: this is a *strength-level* rewrite unless the model class supplies a law of evidence flow itself. The evidence for the conclusion is still obtained by querying the WM state, unless we can prove more.

6 Knuth–Skilling bridges: valuations and regraduation

Knuth–Skilling (K&S) "foundations of inference" axiomatize valuation schemes on lattices. In our setting, K&S plays two roles:

1. **Probability as a valuation/view.** Within Boolean world sets, probabilities arise as valuations and can be handled in standard ways.

2. **Regraduation bridges for computation.** K&S representation theorems justify mapping an abstract valuation to a real scale where combination becomes addition or scaled multiplication, enabling efficient computation.

In the Lean development, a regraduation map

$$\Theta : \alpha \to \mathbb{R}$$

turns an abstract additive valuation algebra into real addition:

$$\Theta(x \star y) = \Theta(x) + \Theta(y).$$

This is explicitly a **scale change**, not probability normalization.

This matters for PLN because:

- it provides a principled bridge between abstract evidence/valuation objects and numerical algorithms,
- it lets the VE engine operate over K&S-valued factor graphs after regrading,
- it keeps "probability" as a *view* rather than the foundational type.

7 Factor graphs in two roles: semantics vs control (MORK alignment)

A key conceptual cleanup:

1. **Semantic factor graphs:** factors *are the model*. They represent a genuine factorization (BN/MRF), and VE / junction tree is exact relative to that model class.

2. **Operational factor graphs:** factors are *rule schemas* or *control edges*. Message passing is a heuristic that schedules which queries to ask / which rewrites to attempt. It becomes *semantics* only after compilation: each operational factor is justified by a semantic factor (or by a proven admissible rewrite under Σ).

7.1 Evidence ledgers and positive feedback loops

Message passing, activation spreading, and positive feedback loops are not disallowed. They just need accountability:

- Track provenance and overlap to avoid double-counting evidence.
- Track Σ hypotheses explicitly (e.g. "this subgraph is Markov").
- Maintain a ledger of approximation error / drift estimates.

This viewpoint matches the "geodesic inference control" intuition: control can be decentralized, but evidence must not be silently duplicated.

7.2 Anchor states and recurrence

Practical systems often revisit certain “anchor” situations (daily routines, infrastructure states, public transit modes). If the agent revisits anchor-like regimes infinitely often, the system can calibrate its approximation ledgers. This suggests an empirical condition:

> *We need enough recurrence in the experienced state space to re-estimate which simplifying assumptions are valid in the domains where we rely on them.*

When recurrence is weak, the model-class validity itself must be treated as uncertain (and updated like any other hypothesis).

8 Gluing logics: projection operators and probabilistic model-class uncertainty

The “complete logic + tractable bubbles” dream is achievable if we resist one temptation: *never silently treat a simplifying assumption as if it were true.*

8.1 Model class as a hypothesis

Let M range over model classes (Markov, Beta–Bernoulli, Gaussian, etc.). Treat “the domain is Markov” as a hypothesis with probability (or evidence) mass. Then:

$$P(q) \;=\; \sum_{M} P(M)\, P(q \mid M),$$

where $P(q \mid M)$ can be computed in a tractable “bubble” (factor graph, linear model, etc.), and $P(M)$ is estimated in the general world-model layer.

8.2 Projections (forgetful maps) as approximation interfaces

Let $\pi_M : \mathsf{State} \to \mathsf{State}_M$ be a projection from a rich state to a restricted state appropriate for model class M. Soundness questions become:

- What queries commute with projection? (exactness of compiled tactics)
- What inequalities or bounds hold? (safe approximations)
- How do we revise $P(M)$ when projections fail diagnostic tests?

This is the correct place for “topological gluing” intuitions: projections are like restriction maps, and consistency conditions correspond to commuting diagrams.

8.3 Provenance and overlap: partial revision instead of wrong addition

Evidence sources can overlap (shared observations, shared causal channels, “data incest”). If overlap is unknown, revision should not be a total operation. A principled fix is:

- Use a **partial** commutative monoid (separation algebra): $W_1 \oplus W_2$ is defined only when the fragments are known independent/disjoint.

- If disjointness is uncertain, either keep fragments separate (delayed fusion), or fuse conservatively using bounds (credal sets / Fréchet bounds / covariance intersection analogues).
- Track provenance IDs so "additivity" is never applied to overlapping sources.

This is the clean mathematical home of "approximation ledgers".

9 Quantifiers in the WM calculus

PLN must support probabilistic reasoning over quantified claims:

$$P(\exists x.\, \mathrm{BlackSwan}(x)), \qquad P(\forall x.\, \mathrm{Human}(x) \Rightarrow \mathrm{Mortal}(x)).$$

9.1 Satisfying sets and weakness: a usable semantics

A clean PLN-compatible semantics treats a predicate as a *frame-valued satisfying set*:

$$\chi_P : U \to \mathsf{Evidence},$$

analogous to a subobject classifier in a topos. Then universal quantification is evaluated via Goertzel's *weakness* on the diagonal relation:

$$\forall x.\, P(x) \quad \rightsquigarrow \quad \mathrm{weakness}_\mu(\{(u,v) \mid P(u) \wedge P(v)\})$$

for an explicit weight function μ. Existential quantification can be defined by De Morgan using Heyting negation:

$$\exists x.\, P(x) := \neg\forall x.\, \neg P(x).$$

Two distinct quantifier views are worth keeping separate:

- **Extensional (classical-ish) quantifiers:** "all individuals satisfy".
- **Typicality quantifiers:** "generality" measured by weakness/diagonal mass.

The separation prevents subtle category mistakes: typicality is not classical $\forall$, but it is exactly what many PLN uses want.

9.2 How quantifiers live in the WM calculus

In the WM architecture, quantified statements are simply queries q: the world model state decides how to compute $\mathsf{evidence}(W, q)$. Different world-model classes yield different quantifier semantics: finite-domain exact, sampled approximate, or quasi-Borel/infinite.

Fast quantified rules are again Σ-guarded rewrites (e.g. Skolemization, instantiation heuristics), with explicit approximation tracking.

9.3 Higher-order quantification and higher-order probability

Quantifying over predicates (HO) is hard in crisp logics because the domain is huge. But probabilistic semantics changes the game:

- **Higher-order probabilities can be flattened.** Kyburg argues that "second-order probabilities" add no conceptual power: they can be represented as ordinary joint distributions over expanded spaces. So hyperpriors/hyperparameters are *first-order* variables in a larger model.
- **Henkin semantics as a pragmatic HO target.** Instead of full set-theoretic HO semantics, quantify only over a chosen collection of predicates. This yields completeness theorems similar to first-order logic, and matches "representable concept" quantification.
- **Quasi-Borel / measurable HO as the long game.** If the WM state is a probabilistic object on a QBS-like domain, HO quantification can be interpreted as integration over function spaces—still difficult, but at least mathematically well-defined.

10 Temporal/fixpoint reasoning: modal μ-calculus as a PLN bridge (optional)

Once reasoning becomes stateful (WM revision) and iterative (message passing), temporal/fixpoint logics become natural. One promising bridge is the modal μ-calculus:

- temporal operators ("eventually", "always") can often be presented as least/greatest fixpoints;
- PLN temporal operators can be embedded into μ-calculus under a suitable semantics;
- fixpoint structure also aligns with recurrent "anchor states" and invariant tracking.

In practice: keep temporal/fixpoint reasoning as an additional query language and add rewrite rules that are sound relative to the chosen transition-model WM class.

11 Process calculi, linear logic, and graph-structured semantics

MeTTa is implemented in Rholang (a ρ-calculus descendant), and the project involves experts in higher-order process calculi and categorical semantics. This suggests a productive alignment:

- WM states can be treated as *resources* (linear logic flavor).
- Revision is a commutative monoidal "addition" of evidence/resources.
- Query answering is a form of *observation* (a morphism out of state).
- Operational factor graphs become explicit *process networks* that schedule observations and rewrites, while the WM semantics provides correctness conditions.

This is a plausible bridge to OSLF / type-theoretic presentations and graph-structured lambda theories: quantifiers become binders, and factor-graph compilation becomes a typed elaboration step.

12 What “sound and complete” should mean here

You want a proof calculus, not only a semantic kernel. The WM calculus is already a proof calculus skeleton; the missing pieces are *fragment-specific* soundness/completeness theorems.

A clean framing:

12.1 Soundness

For a fragment $\mathcal{L}$ and semantics $[\![\cdot]\!]$:

> **Soundness:** every derivable judgment corresponds to a true semantic statement about evidence/strength in the target model class.

In practice:

- Kernel soundness: revision/query laws (commutation, additivity).
- Rewrite soundness: each Σ-guarded compiled rule preserves the chosen view (e.g. strength).
- Approximate soundness: rules preserve *bounds* or carry explicit error budget.

12.2 Completeness (fragmented, but meaningful)

Completeness should be stated relative to a declared semantics and fragment:

- **Propositional fragment:** evidence-quantale semantics vs IPL/intermediate logics.
- **First-order fragment:** satisfying-set/weakness quantifiers (finite or Henkin FO).
- **BN fragment:** completeness relative to BN query answering (VE/junction tree).
- **Higher-order fragment:** Henkin completeness; QBS semantics as aspirational.

This is not a retreat. It is how serious proof theory works: different fragments have different theorems.

13 Lean artifact map (for posterity)

These are the main artifacts referenced by the architecture (names may evolve, but the roles should not):

- `Mettapedia/Logic/PLNWorldModel.lean`: WM kernel (states, revision, query extraction).
- `Mettapedia/Logic/PLNWorldModelCalculus.lean`: Σ-guarded rewrite-rule schema and explicit strength-view judgments.
- `Mettapedia/Logic/PLNJointEvidence.lean`: complete joint-evidence (Dirichlet-over-worlds) WM instance.
- `Mettapedia/Logic/PLNJointEvidenceNoGo.lean`: formal no-go theorem for local link-evidence propagation.
- `Mettapedia/Logic/EvidenceQuantale.lean`: algebraic evidence layer (quantale/Heyting structure).

- `Mettapedia/Logic/PLNBNCompilation.lean`: Σ-guarded screening-off rewrites for BN sublayer.
- `Mettapedia/ProbabilityTheory/BayesianNetworks/FactorGraph.lean`: factor graph semantics.
- `Mettapedia/ProbabilityTheory/BayesianNetworks/VariableElimination.lean`: VE query engine (sum-of-products).
- `Mettapedia/ProbabilityTheory/KnuthSkilling/Bridges/ValuationAlgebra.lean`: K&S regraduation bridge.
- `Mettapedia/Logic/PLNFirstOrder/QuantifierSemantics.lean`: extensional vs typicality quantifier views.

14 Roadmap: "final steps" that actually matter

The most valuable next theorems (because they make PLN usable and honest):

1. **Independence-to-screening-off bridge.** Prove that d-separation / conditional independence implies the screening-off equalities required by PLN deduction rules (including both B and $\neg B$ branches), under explicit positivity.
2. **VE correctness as a WM-strength rule.** Show that VE computes the same strength view as querying the BN world model, so compiled fast rules can target $\vdash_s$ judgments.
3. **Quantifier rewrite rules with explicit approximation ledgers.** Hook satisfying-set quantifiers into the rewrite layer, with soundness under stated domain hypotheses (finite, sampled, exchangeable, etc.).
4. **Fragment completeness statements.** Prove completeness for the propositional and (selected) FO/Henkin fragments relative to the semantics used in code. Treat QBS HO as a separate long-horizon milestone.
5. **Operational integration (MORK).** Implement message passing / activation spreading as inference control that proposes Σ-guarded rewrites, while the WM layer remains the single source of semantic truth.

Acknowledgments / provenance

This note compiles ideas emerging from ongoing collaboration around PLN, MeTTa, and the Lean formalization in `Mettapedia`, with conceptual inspiration from: Knuth–Skilling foundations of inference, valuation algebras, belief propagation / junction trees, topos semantics of quantifiers, and higher-order process calculi.

References

[1] K. H. Knuth and J. Skilling. *Foundations of Inference.* (K&S representation theorems for valuations on lattices.)

[2] J. Kohlas and P. Shenoy. Computation in valuation algebras. (Axioms: combination $\otimes$ and marginalization $\downarrow$; local computation.)

[3] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. (Variable elimination; complexity via treewidth.)

[4] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* (Bayesian networks, d-separation.)

[5] H. E. Kyburg. Higher order probabilities and interval probabilities. (Flattening higher-order probabilities into joint distributions.)

[6] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic.* (Subobject classifiers, semantics for quantifiers.)