T5 Cost Uniqueness and the Certificate Circle

What Completing “T5” Certifies in the `reality` Repository

December 20, 2025

Abstract

This note explains the mathematical and engineering content of completing “T5” in the `reality` repository’s Lean formalization workflow. In this codebase, T5 is packaged as a *certificate* (`IndisputableMonolith/Verification/T5UniqueCert.lean`) asserting a uniqueness theorem for the Recognition Science cost function J: any function satisfying the certified “JensenSketch” obligations agrees with J on all positive reals. We describe (i) the certificate-circle methodology used to expand the machine-checked surface while enforcing non-circularity, (ii) the explicit definition and key identities of J, (iii) the precise statement of the certified T5 theorem, and (iv) what this theorem does and does not guarantee about the overall theory stack.

1 The certificate circle: what it is and why it matters

The repository uses a disciplined pattern to grow what it calls the *certified surface*: the set of claims that are not merely written down, but *machine-checked* and imported by a top-level certificate module.

Definition 1 (Certificate module (repo convention)). *A certificate module is a Lean file that defines a record `...Cert` together with:*

- *a predicate `...Cert.verified : Prop` that expresses the intended claim; and*
- *a theorem `...Cert.verified_any : ...Cert.verified` proved with no `sorry` and no vacuous placeholders.*

In practice, the record often carries no data; its role is to package a named proposition and a named proof in a stable import path.

This pattern supports a “certificate circle”: each completed step enlarges the certified surface, and future steps are required to import and build on prior certificates rather than reintroducing informal assumptions.

1.1 Where T5 sits in the certificate chain

The current top-level certificate anchor is `IndisputableMonolith/URCGenerators/UltimateCPMClosureCert.lea` This module imports the audit certificate `IndisputableMonolith/Verification/NonCircularityCert.lean`. Since T5 is a conjunct of `NonCircularityCert.verified`, completing T5 places the T5 theorem inside the transitive import closure of the top-level certificate.

1.2 Non-circularity as an explicit invariant

The same workflow imposes a crucial constraint: *non-circularity*. The repo explicitly audits against smuggling patterns such as:

- proving "matches" by hard-coding empirical numerals and using `rfl`;
- `True` stubs or vacuous existentials ($\exists c$, `True`);
- hidden axioms introduced via typeclass instances.

At the Lean level, the main audit anchor is `IndisputableMonolith/Verification/NonCircularityCert.lean` which is a large conjunction of certified facts (defaults, invariances, bridges, cost facts, etc.). T5 appears as one conjunct in this audit certificate, meaning it is explicitly part of the "what is certified" story for the project.

2 The cost function J in the formalization

The Recognition Science cost used throughout the certified surface is the function $J : \mathbb{R}_{>0} \to \mathbb{R}$ defined in Lean as `Jcost`. Its definition is explicit and algebraic:

Definition 2 (J-cost)**.** *For $x \in \mathbb{R}$, define*

$$J(x) \;:=\; \frac{x + x^{-1}}{2} - 1.$$

In the certified statements, the relevant domain is $x > 0$.

Several basic properties are proved in the cost layer. Two identities are especially important:

Remark 1 (Symmetry and squared form)**.** *For $x > 0$,*

$$J(x) = J(x^{-1}), \qquad J(x) = \frac{(x-1)^2}{2x} \geq 0.$$

The squared form makes non-negativity and the "unique minimum at $x = 1$" intuition immediate.

2.1 Log-coordinates and the hyperbolic connection

The cost layer also defines a log-coordinate representation

$$J_{\log}(t) \;:=\; J(e^t).$$

In Lean, this is `Jlog`. A key certified identity is:

$$J_{\log}(t) = \cosh(t) - 1.$$

This ties the cost geometry to a canonical hyperbolic function and supports analytic statements about derivatives, convexity, and normalization (e.g. $J''_{\log}(0) = 1$).

3 The T5 theorem that was completed

3.1 The certified interface: `JensenSketch`

In the codebase, T5 is phrased as a uniqueness result relative to a small interface called `JensenSketch`. Informally, `JensenSketch F` asserts that F is:

- symmetric under inversion ($F(x) = F(1/x)$ for $x > 0$);
- normalized at 1 ($F(1) = 0$);
- "squeezed" on the exponential axis by J, i.e. for all $t \in \mathbb{R}$, $F(e^t) \le J(e^t)$ and $J(e^t) \le F(e^t)$.

The squeeze is logically equivalent to the pointwise equality $F(e^t) = J(e^t)$, but it is written as two inequalities to match how such equalities are typically obtained in analysis (upper and lower bounds derived independently).

3.2 T5 (as certified)

The certificate `IndisputableMonolith/Verification/T5UniqueCert.lean` packages the following statement:

Theorem 1 (T5 cost uniqueness on $\mathbb{R}_{>0}$)**.** *Let $F : \mathbb{R} \to \mathbb{R}$. If F satisfies the `JensenSketch` obligations, then for all $x > 0$,*

$$F(x) = J(x).$$

Remark 2 (Proof idea)**.** *The proof is conceptually simple: the `JensenSketch` bounds give equality on the exponential axis, and every $x > 0$ can be written as $x = e^{\log x}$, so equality transfers from $\{e^t : t \in \mathbb{R}\}$ to all of $\mathbb{R}_{>0}$. In Lean this is implemented as `T5_cost_uniqueness_on_pos`.*

4 A "full" (classical) uniqueness route exists in the repo

While the certified T5 theorem is phrased at the `JensenSketch` interface, the repository also contains a more classical uniqueness theorem (`IndisputableMonolith/CostUniqueness.lean`). That theorem shows how one can force $F = J$ from more recognizable analytic hypotheses such as: symmetry, strict convexity on $\mathbb{R}_{>0}$, calibration in log-coordinates, continuity, and a cosh-type functional identity.

This matters for interpretation:

- The T5 certificate says: *once the JensenSketch obligations are met*, the cost is forced.
- The deeper uniqueness theorem suggests: there are principled analytic roads to proving those obligations without hard-coding J into the assumptions.

5 What completing T5 means for the certificate circle

Completing T5 has two distinct meanings: a mathematical meaning and a certification meaning.

5.1 Mathematical meaning

Mathematically, T5 asserts that J is not an arbitrary choice once the project commits to the invariances and normalizations encoded by `JensenSketch`. Downstream theorems that depend on these obligations cannot silently "swap in" a different cost function to fit desired outcomes: the interface forces the same J everywhere on $\mathbb{R}_{>0}$.

5.2 Certification meaning

Engineering-wise, T5 is now part of the auditable, import-stable certified surface:

- `T5UniqueCert` provides a named proposition (`verified`) and a named proof (`verified_any`) with no `sorry`;
- `NonCircularityCert` imports `T5UniqueCert` and includes it as one conjunct in its verified predicate, so the audit certificate explicitly depends on T5.

In other words: T5 is not only proven somewhere in the library; it is a *declared checkpoint* in the repo's audit certificate, and thus a stable part of what the project is willing to call "certified".

6 What T5 does *not* claim

It is equally important to state the boundary:

- T5 does not claim that empirical physics selects J from nothing; it claims uniqueness *conditional on* the stated obligations.
- T5 does not certify any CODATA/SI numerics. The repo treats hard-coded empirical numerics as quarantined from the certified surface.
- T5 does not automatically certify every downstream "closure" claim; it is one component in a broader non-circularity story that also includes explicit match evaluators, calibration witnesses, and rescaling invariances.

7 Reproducibility (artifact map)

The relevant Lean artifacts are:

- `IndisputableMonolith/URCGenerators/UltimateCPMClosureCert.lean` (top-level certificate anchor that imports `NonCircularityCert`);
- `IndisputableMonolith/Cost.lean` (definition of `Jcost`, `Jlog`, and `JensenSketch`, plus `T5_cost_uniqueness_on_pos`);
- `IndisputableMonolith/Verification/T5UniqueCert.lean` (packages T5 as a cert);
- `IndisputableMonolith/Verification/NonCircularityCert.lean` (imports and asserts T5 as part of the audit certificate);
- `IndisputableMonolith/CostUniqueness.lean` (a stronger, explicit-hypothesis uniqueness theorem that explains one non-circular route to deriving J).

Typical local verification commands (from repo root) are of the form:

```
lake build IndisputableMonolith.URCGenerators.UltimateCPMClosureCert
lake build IndisputableMonolith.Verification.T5UniqueCert
lake build IndisputableMonolith.Verification.NonCircularityCert
```