Knuth-Skilling Formalization
Review Walkthrough

Codex 5.2, Claude 4.5, Zar Goertzel

January 21, 2026

Abstract

This document provides a systematic walkthrough of the Lean 4 formalization of Knuth & Skilling's "Foundations of Inference" (2012). Each section lists the main theorem statements with their Lean proofs and file locations, organized by the corresponding K&S paper sections. The formalization is **complete with zero sorries** in all core files.

Quickstart for Reviewers

- **High-level view:** Start with Section 12 (Summary) for the consolidated theorem list and key discoveries.
- **Audit Appendix A (sum rule):** See Section 5, then the three proof paths (Hölder / Grid / Direct Cuts).
- **Audit Appendix B (product rule):** See Section 6 for the two independent proof paths.
- **Audit Appendix C (entropy / variational theorem):** See Section 7 and the counterexamples in Section 11.
- **Build commands:** See Section 13 (Build Instructions) at the end.

Contents

1 Overview and File Structure

1.1 K&S Paper Coverage

K&S Section	Topic	Lean Files	Status
Sections 1–2	Sum-side Axioms (Sym 0–2)	`Basic.lean`, `Algebra.lean`	Complete
Section 3	Probability	`SymmetricalFoundation.lean`	Complete
Section 4	Quantum theory	`SymmetricalFoundation.lean`, `TwoDimClassification.lean`	Complete
Section 6	Divergence	`Divergence.lean`	Complete
Section 7	Conditional prob.	`ConditionalProbability/Basic.lean`	Complete
Section 8	Info/Entropy	`InformationEntropy.lean`	Complete
Appendix A	Representation	`RepresentationTheorem/Main.lean`	Complete
Appendix B	Product (Sym 3–4)	`ProductTheorem/Main.lean`	Complete
Appendix C	Variational	`VariationalTheorem.lean`	Complete

1.2 Key Files

File Path	Contents
`.../KnuthSkilling.lean`	Main entrypoint (imports all)
`.../KnuthSkilling/Basic.lean`	Core: `KnuthSkillingAlgebraBase` (Sym 0–2)
`.../KnuthSkilling/Algebra.lean`	`iterate_op`, `KSSeparation`
`.../Separation/SandwichSeparation.lean`	Archimedean + Commutativity
`.../Separation/HolderEmbedding.lean`	Identity-free representation (Hölder path)
`.../RepresentationTheorem/Main.lean`	Appendix A theorem (public API)
`.../ProductTheorem/Main.lean`	Appendix B: K&S path (via Appendix A)
`.../ProductTheorem/Alternative/DirectProof.lean`	Appendix B: Direct algebraic path
`.../ProductTheorem/ScaledMultRep.lean`	Appendix B: Common interface
`.../VariationalTheorem.lean`	Appendix C entropy
`.../SymmetricalFoundation.lean`	Section 4 quantum theory
`.../TwoDimClassification.lean`	2D algebra classification
`.../Divergence.lean`	Section 6 divergence
`.../InformationEntropy.lean`	Section 8 entropy
`.../ConditionalProbability/Basic.lean`	Section 7 probability calculus

All paths are relative to the Mettapedia project root directory.

2 Core Sum-Side Axioms (K&S Sections 1–2)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Basic.lean`

2.1 K&S Sum-Side Symmetries (0–2)

K&S present symmetries in two groups. The **sum-side** symmetries (0–2) govern the combination operation $\oplus$:

- **Symmetry 0** (Fidelity): $\bar{x} < \bar{y} \Rightarrow x < y$
- **Symmetry 1** (Monotonicity): $\bar{x} < \bar{y} \Rightarrow \bar{x} \oplus \bar{z} < \bar{y} \oplus \bar{z}$
- **Symmetry 2** (Associativity): $(\bar{x} \oplus \bar{y}) \oplus \bar{z} = \bar{x} \oplus (\bar{y} \oplus \bar{z})$

The **product-side** symmetries (3–4) are formalized separately in `ProductTheorem/Main.lean` (see Section 6).

Definition 2.1 (`KSSemigroupBase`)**. Lines 148–156, Basic.lean**
Identity-free core structure containing only Symmetries 0–2.

Listing 1: Identity-free semigroup base

```
class KSSemigroupBase (alpha : Type*) extends LinearOrder alpha where
  op : alpha → alpha → alpha                          -- combination operation
  op_assoc : ∀ x y z : alpha, op (op x y) z = op x (op y z)   -- Sym 2
  op_strictMono_left : ∀ y : alpha, StrictMono (fun x => op x y)    -- Sym 0+1
  op_strictMono_right : ∀ x : alpha, StrictMono (fun y => op x y)   -- Sym 0+1
```

Definition 2.2 (`KnuthSkillingAlgebraBase`)**. Lines 172–180, Basic.lean**
Extends `KSSemigroupBase` with identity element and positivity.

Listing 2: Core K&S algebra structure (extends semigroup base)

```
class KnuthSkillingAlgebraBase (alpha : Type*) extends KSSemigroupBase alpha where
  ident : alpha                                         -- identity element
  op_ident_right : ∀ x : alpha, op x ident = x
  op_ident_left : ∀ x : alpha, op ident x = x
  ident_le : ∀ x : alpha, ident ≤ x            -- positivity
```

Remark 2.3 (Implicit Linear Order). K&S never explicitly state that elements are totally ordered, but their proofs rely on trichotomy. We make this explicit via `LinearOrder`.

Remark 2.4 (Identity Element—NOT in K&S Axioms). **Important**: The identity element (`ident`) is **not** among K&S's numbered symmetries. K&S explicitly state that the bottom element $\perp$ is **optional**:

> "with the bottom element optional" (K&S line 320)
> "Some mathematicians opt to include the bottom element on aesthetic grounds, whereas others opt to exclude it" (K&S lines 340–341)

Our formalization **proves** K&S's claim via the `KSSemigroupBase` hierarchy:

- **Identity-free path** (`HolderEmbedding.lean:296--302`): `representation_semigroup` proves the representation theorem without identity, giving $\Theta : \alpha \to \mathbb{R}$ with order-preservation and additivity.
- **With identity** (`HolderEmbedding.lean:309--312`): `identity_gives_canonical_normalization` shows identity provides **canonical normalization**: $\Theta(\text{ident}) = 0$. Without identity, Θ is defined only up to an additive constant.

Architectural hierarchy:

$$\texttt{KSSemigroupBase}\ \text{(identity-free)} \xrightarrow{\text{extends}} \texttt{KnuthSkillingAlgebraBase}\ \text{(with identity)}$$

Most of the formalization uses `KnuthSkillingAlgebraBase` for historical reasons—the development was done with identity first—as well as convenience (iteration from $n = 0$, canonical zero point), but the core representation theorem is proven at the identity-free `KSSemigroupBase` level.

Remark 2.5 (Unbundled Axiom Predicates). **Lines 143–166, Basic.lean**

In addition to the bundled typeclasses, we provide **unbundled predicates** for each axiom. This enables flexible hypothesis tracking—use individual predicates when you need minimal assumptions, or bundled classes when you want ergonomic access to multiple axioms.

Sum-side predicates (Basic.lean):

- `OpAssoc op` (line 143): $\forall x\, y\, z,\ \mathrm{op}(\mathrm{op}(x, y), z) = \mathrm{op}(x, \mathrm{op}(y, z))$
- `OpStrictMonoLeft op` (line 147): $\forall y,\ \mathrm{StrictMono}(\lambda x.\, \mathrm{op}(x, y))$
- `OpStrictMonoRight op` (line 151): $\forall x,\ \mathrm{StrictMono}(\lambda y.\, \mathrm{op}(x, y))$
- `OpIdentLeft op e` (line 155): $\forall x,\ \mathrm{op}(e, x) = x$
- `OpIdentRight op e` (line 159): $\forall x,\ \mathrm{op}(x, e) = x$
- `IdentIsMin e` (line 163): $\forall x,\ e \le x$

Connection theorems: The `KSSemigroupBase` and `KnuthSkillingAlgebraBase` namespaces provide lemmas like `KSSemigroupBase.opAssoc` that extract the unbundled predicate from a bundled instance.

2.2 Iteration

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Algebra.lean`

Definition 2.6 (`iterate_op`)**. Lines 23–25, Algebra.lean**

K&S Paper Reference: This corresponds to K&S's use of "n copies of x" in their proofs, written as x^n or nx depending on context (see K&S equations around line 370–380 in the TeX source). The iteration builds repeated applications of $\oplus$: $x^n = \underbrace{x \oplus x \oplus \cdots \oplus x}_{n\ \text{times}}$.

Listing 3: Iteration definition

```
def iterate_op (x : alpha) : N → alpha
  | 0 => ident
  | n + 1 => op x (iterate_op x n)
```

This builds the sequence: $\mathrm{ident}, x, x \oplus x, x \oplus (x \oplus x), \ldots$

2.3 Identity-Free Iteration

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Basic.lean`

Definition 2.7 (`iterate_op_pnat`)**. Lines 305–306, Basic.lean**

For identity-free reasoning, we define iteration using positive natural numbers ($\mathbb{N}^+$) instead of $\mathbb{N}$. This works on the weaker `KSSemigroupBase` (no identity required).

Listing 4: Identity-free iteration

```
def iterate_op_pnat [KSSemigroupBase alpha] (x : alpha) (n : ℕ+) : alpha :=
  iterate_op_pnat_aux x (n.val - 1)

private def iterate_op_pnat_aux (x : alpha) : ℕ → alpha
  | 0 => x          -- n=0 maps to x^1 = x
  | n + 1 => op x (iterate_op_pnat_aux x n)
```

Remark 2.8 (Key Properties).

- `iterate_op_pnat x 1 = x` (base case is x, not ident)
- `iterate_op_pnat x (n+1) = op x (iterate_op_pnat x n)` (recursion)
- Builds the sequence: $x, x \oplus x, x \oplus (x \oplus x), \ldots$ (no identity!)
- **Connection**: `iterate_op_pnat x n = iterate_op x n.val` when $n \geq 1$ (`Algebra.lean:198`)

Usage: This identity-free version is used in:

- `KSSeparationPNat` (separation axioms using $\mathbb{N}^+$ iteration)
- `Separation/HolderEmbedding.lean` (identity-free representation path)
- `RepresentationTheorem/Alternative/DirectCuts.lean` (alternative representation using cuts)

3 The Separation Property

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Algebra.lean`

Definition 3.1 (`KSSeparationSemigroup`)**. Lines 292–297, Algebra.lean**

The separation property allows "sandwiching" any pair of distinct positive elements using powers of a base element. Uses `IsPositive` (element increases everything) instead of `ident < a`, and `iterate_op_pnat` ($\mathbb{N}^+$ iteration) instead of `iterate_op` ($\mathbb{N}$ iteration).

Listing 5: Separation axiom (identity-free)

```
class KSSeparationSemigroup (alpha : Type*) [KSSemigroupBase alpha] where
  separation : ∀ {a x y : alpha}, IsPositive a → IsPositive x → IsPositive y → x < y →
    ∃ n m : ℕ+, iterate_op_pnat x m < iterate_op_pnat a n /\
                iterate_op_pnat a n ≤ iterate_op_pnat y m
```

Intuition: For any positive base a and distinct $x < y$, we can find exponents $(n, m) \in \mathbb{N}^+$ such that $x^m < a^n \leq y^m$.

Key advantage: Works on `KSSemigroupBase` (no identity required).

Remark 3.2 (Unbundled Separation Predicates). **Lines 285–380, Algebra.lean**

Following the unified axiom organization, separation also has unbundled predicates:

- `SeparationSemigroupProp` (line 285): Identity-free separation (uses $\mathbb{N}^+$)
- `SeparationSemigroupStrictProp` (line 293): Strict variant with $<$ on both sides
- `SeparationProp` (line 370): With identity (uses $\mathbb{N}$)

- `SeparationStrictProp` (line 380): Strict variant with identity

Connection theorems: The `KSSeparation` and `KSSeparationSemigroup` namespaces provide extraction lemmas (e.g., `KSSeparationSemigroup.separationSemigroupProp`) and convenience wrappers.

4 Derivable Consequences

4.1 Archimedean Property

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/RepresentationTheorem/Alternative/DirectCuts.le`

Theorem 4.1 (archimedean_pnat_of_separation)**.** ***Lines 1185–1210, DirectCuts.lean***
Under `KSSeparationSemigroup`, for any positive a and x, there exists $N \in \mathbb{N}^+$ such that $a \le x^N$.

Listing 6: Archimedean from Separation (identity-free)

```
theorem archimedean_pnat_of_separation
    [KSSemigroupBase alpha] [KSSeparationSemigroup alpha]
    (a x : alpha) (ha : IsPositive a) (hx : IsPositive x) :
    ∃ N : N+, a ≤ iterate_op_pnat x N := by
  -- Case 1: a ≤ x (trivial)
  by_cases hax : a ≤ x
  . exact <1, by simpa only [iterate_op_pnat_one] using hax>
  -- Case 2: x < a
  -- Use separation with base x on interval (a, op a a)
  have ha2_lt : a < op a a := ha a
  obtain <n, m, h_lower, h_upper> := KSSeparationSemigroup.sep hx ha ha2_pos
      ha2_lt
  -- From a^m < x^n: conclude a ≤ x^n
  use n
  have ha_le_am : a ≤ iterate_op_pnat a m := by
    rw [← iterate_op_pnat_one a]
    exact iterate_op_pnat_mono a ha (PN.one_le m)
  exact le_of_lt (lt_of_le_of_lt ha_le_am h_lower)
```

4.2 Commutativity via NoAnomalousPairs (Main Route)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Separation/AnomalousPairs.lean`
The **main proof route** uses the classical Hölder/Alimov/Fuchs theorem from ordered semigroup theory:

Definition 4.2 (`NoAnomalousPairs`)**.** An ordered semigroup has no anomalous pairs if no two elements a, b have iterates "squeezed" forever: $a^n < b^n < a^{n+1}$ for all n.

Theorem 4.3 (Hölder 1901, Alimov 1950)**.** *In a linearly ordered cancellative semigroup, the absence of anomalous pairs is necessary and sufficient for an additive embedding into* $(\mathbb{R}, +)$.

Key implication chain:

1. **Separation $\Rightarrow$ No Anomalous Pairs**: The separation property provides witnesses that break any potential squeeze (formalized in `Separation/AnomalousPairs.lean`).

2. **No Anomalous Pairs ⇒ Hölder Embedding**: Classical result (Hölder 1901, Alimov 1950, Fuchs 1963), formalized by Eric Luap in `OrderedSemigroups`.

3. **Hölder Embedding ⇒ Commutativity**: The embedding into $(\mathbb{R}, +)$ forces commutativity.

4. **Real Representation ⇒ Separation**: Rational density in $\mathbb{R}$ provides the sandwich witnesses.

Equivalence: Under our standing hypotheses (linearly ordered, cancellative, associative, strictly monotone, identity as minimum), these are equivalent:

$$\text{Separation} \iff \text{No Anomalous Pairs} \iff \text{Additive real representation}$$

Commutativity is a *derived property*, not an independent axiom. This makes "no anomalous pairs" a **classical sharp Archimedean-type condition** in the standard setting of linearly ordered cancellative semigroups.

References:

- Hölder, O. (1901). "Die Axiome der Quantität und die Lehre vom Mass." *Ber. Verh. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Cl.* 53, 1–64.
- Alimov, N. G. (1950). "On ordered semigroups." *Izv. Akad. Nauk SSSR Ser. Mat.* 14, 569–576.
- Fuchs, L. (1963). *Partially Ordered Algebraic Systems.* Pergamon Press.
- Luap, E. (2024). "OrderedSemigroups: Formalization of Ordered Semigroups in Lean 4."

Remark 4.4 (Alternative: Direct Mass Counting). There is also a direct "mass counting" proof (`SandwichSeparation.lean:438--491`): Assume $(x \oplus y) < (y \oplus x)$. Apply separation to get $(x \oplus y)^m < (x \oplus y)^n \leq (y \oplus x)^m$. By associativity, both $(x \oplus y)^n$ and $(y \oplus x)^m$ contain the same multiset of atoms, so $(x \oplus y)^n > (y \oplus x)^m$ when $n > m$—contradiction.

Listing 7: Key mass counting lemma

```
theorem xy_pow_gt_yx_pow (x y : alpha) (hx : ident < x) (hy : ident < y)
    (n m : N) (hm : m ≥ 1) (hnm : n > m) :
    iterate_op (op x y) n > iterate_op (op y x) m := by
  -- Uses associativity to show more copies of positive atoms = larger
  ...
```

5 The Representation Theorem (Appendix A)

Files:

- `RepresentationTheorem/Main.lean` (public API)
- `RepresentationTheorem/Globalization.lean` (850 lines: globalization proof using Core machinery)
- `RepresentationTheorem/Core/` (∼17,700 lines: grid infrastructure)
  - `Core/MultiGrid.lean`: `AtomFamily`, `MultiGridRep`, grid representations

– `Core/Induction/`: Inductive extension theorems (`Construction`, `ThetaPrime`, `DeltaShift`)
– `Core/OneDimensional.lean`: Base case (single atom)
– `Core/Prelude.lean`: Foundational lemmas

Architecture: `Globalization.lean` imports `Core.All` and orchestrates the grid machinery to prove global representation.

5.1 Three Independent Proof Paths

The formalization provides **three complete, independent proof routes** to the representation theorem:

1. **Hölder embedding** (MAIN ROUTE, weakest assumptions): Uses NoAnomalousPairs condition, classical ordered semigroup theory (Hölder 1901, Alimov 1950, Fuchs 1963), formalized via Eric Luap's `OrderedSemigroups`.
2. **Dedekind cuts** (alternative): Uses Separation property with Hölder/Dedekind cuts construction, bypassing the grid machinery.
3. **Grid induction** (K&S-style): Uses multi-dimensional grid representations and induction on atom families, following K&S's original approach.

Historical development:

- First: Grid/induction path (following K&S's original approach)
- Second: Cuts path discovered by Claude Code as superior
- Third: Hölder/Alimov path discovered by GPT-5.2 Pro as the best (weakest assumptions, classical connection)

All three proofs are complete with zero sorries. The Hölder path uses **NoAnomalousPairs only**—the weakest known condition sufficient for the representation theorem.

5.2 Proof Architecture 1: The Hölder Path (Main Route)

File: `Separation/HolderEmbedding.lean`

The Hölder path is the **recommended main route** because it uses the weakest hypotheses and connects to classical ordered semigroup theory.

Theorem 5.1 (holder_embedding_of_noAnomalousPairs)**.** ***Lines 182–185, HolderEmbedding.lean***

If a K&S algebra has no anomalous pairs, it embeds into $\mathbb{R}$*.*

Listing 8: Hölder embedding theorem

```
theorem holder_embedding_of_noAnomalousPairs [NoAnomalousPairs alpha] :
    ∃ G : Subsemigroup (Multiplicative R), Nonempty (alpha =(equiv)*o G) := by
  have h : ~has_anomalous_pair (alpha := alpha) := noAnomalousPairs_iff_eric
  exact holder_not_anom h
```

Theorem 5.2 (`representation_semigroup`). ***Lines 297–303, HolderEmbedding.lean***

Identity-free representation*: NoAnomalousPairs implies additive embedding into $\mathbb{R}$. Θ is defined up to an additive constant (no canonical zero point).*

Listing 9: Identity-free representation theorem

```
theorem representation_semigroup [NoAnomalousPairs alpha] :
    ∃ Theta : alpha → R,
      (∀ a b : alpha, a ≤ b ↔ Theta a ≤ Theta b) /\
      (∀ x y : alpha, Theta (op x y) = Theta x + Theta y) := by
  obtain <G, <iso>> := holder_embedding_of_noAnomalousPairs (alpha := alpha)
  use theta_from_embedding G iso
  exact <theta_preserves_order G iso, theta_additive G iso>
```

Theorem 5.3 (`representation_from_noAnomalousPairs`). ***Lines 269–276, HolderEmbedding.lean***

With identity*: NoAnomalousPairs implies the full representation with $\Theta(ident) = 0$.*

Listing 10: Representation with identity normalization

```
theorem representation_from_noAnomalousPairs [NoAnomalousPairs alpha] :
    ∃ Theta : alpha → R,
      (∀ a b : alpha, a ≤ b ↔ Theta a ≤ Theta b) /\
      Theta ident = 0 /\
      ∀ x y : alpha, Theta (op x y) = Theta x + Theta y := by
  obtain <G, <iso>> := holder_embedding_of_noAnomalousPairs (alpha := alpha)
  use theta_from_embedding G iso
  exact <theta_preserves_order G iso, theta_ident G iso, theta_additive G iso>
```

Key advantage: This path works on `KSSemigroupBase` (identity-free) via `representation_semigroup`, and on `KnuthSkillingAlgebraBase` (with identity) via `representation_from_noAnomalousPairs`.

5.3 Proof Architecture 2: The Grid/Induction Path

The grid-based proof is packaged as the typeclass `RepresentationGlobalization`, which is automatically instantiated when `[KSSeparationStrict` α`]` is available.

Definition 5.4 (`RepresentationGlobalization`). **Lines 54–60, Globalization.lean**

A typeclass packaging the existence of Θ.

Listing 11: RepresentationGlobalization typeclass

```
class RepresentationGlobalization (alpha : Type*)
    [KnuthSkillingAlgebra alpha] [KSSeparation alpha] : Prop where
  ∃_Theta :
    ∃ Theta : alpha → R,
      (∀ a b : alpha, a ≤ b ↔ Theta a ≤ Theta b) /\
      Theta ident = 0 /\
      ∀ x y : alpha, Theta (op x y) = Theta x + Theta y
```

5.3.1 The Globalization Construction ("Triple Family Trick")

The instance `representationGlobalization_of_KSSeparationStrict` (lines 93–850, Globalization.lean) constructs Θ globally using a multi-step process:

1. **Reference atom**: Choose any $a_0 >$ ident as a fixed reference point.

2. **2-atom families**: For each $x >$ ident, build a 2-atom family $F_2 = \{a_0, x\}$ with a `MultiGridRep` R_2 (via `extend_grid_rep_with_atom_of_KSSeparationStrict` from `Core/`).

3. **Define** $\Theta(x)$: Extract the representation value from the grid:

$$\Theta(x) := R_2.\text{Theta\_grid}(\langle x, \text{membership\_proof}\rangle)$$

4. **Well-definedness**: Use 3-atom families $F_3 = \{a_0, a_1, x\}$ to show that $\Theta(x)$ does not depend on the choice of reference atom. Path independence follows from `DeltaSpec_unique` (line 755, `Core/Induction/Construction.lean`).

5. **Order preservation**: For $a < b$, build $F_3 = \{a_0, a, b\}$ and use `MultiGridRep.strictMono` to show $\Theta(a) < \Theta(b)$.

6. **Additivity**: For $x \oplus y$, build $F_3 = \{a_0, x, y\}$ and verify $\Theta(x \oplus y) = \Theta(x) + \Theta(y)$ by path independence across different extension orderings.

Remark 5.5 (Why "Triple Family Trick"?). The name comes from using 3-atom families to mediate between different 2-atom constructions. This technique ensures global consistency: any two definitions of $\Theta(x)$ via different reference atoms must agree, because they both embed into a common 3-atom grid representation.

Remark 5.6 (Identity-Free Grid Infrastructure). The grid construction has **parametric versions** that could work without identity:

- `mu_param F r base`: Grid valuation with explicit base element instead of `ident`
- `kGrid_param F base`: Grid set using `mu_param`
- `mu_pnat`, `kGrid_pnat`: Truly identity-free using $\mathbb{N}^+$ iteration (no 0 exponents)
- `RepresentationGlobalizationAnchor`: Class for representations normalizing to an arbitrary anchor

Currently, the globalization instance uses identity (`representationGlobalization_of_KSSeparationStrict`). An identity-free instance using the parametric infrastructure is marked as **future work** in `Globalization.lean`.

For identity-free representations **today**, use the Hölder path (`HolderEmbedding.lean`) which produces `RepresentationResult` (order + additivity, no normalization constraint).

5.4 Main Theorem Statement

Theorem 5.7 (`associativity_representation`)**.** ***Lines 32–38, RepresentationTheorem/-Main.lean***

K&S Appendix A Main Theorem*: There exists an order embedding $\Theta : \alpha \to \mathbb{R}$ such that:*

1. *Order preservation:* $a \le b \Leftrightarrow \Theta(a) \le \Theta(b)$

2. $\Theta(\mathit{ident}) = 0$

3. *Additivity:* $\Theta(\mathit{op}\ x\ y) = \Theta(x) + \Theta(y)$

Listing 12: Appendix A Representation Theorem (public API)

```
theorem associativity_representation
    (alpha : Type*) [KnuthSkillingAlgebra alpha] [KSSeparation alpha]
    [RepresentationGlobalization alpha] :
    ∃ Theta : alpha → R,
      (∀ a b : alpha, a ≤ b ↔ Theta a ≤ Theta b) /\
      Theta ident = 0 /\
      ∀ x y : alpha, Theta (op x y) = Theta x + Theta y := by
  exact RepresentationGlobalization.∃_Theta (alpha := alpha)
```

Remark 5.8 (Proof Delegation). The theorem statement simply extracts `exists_Theta` from the typeclass. All the actual work happens in the instance construction:

`representationGlobalization_of_KSSeparationStrict` (lines 93–850, Globalization.lean)

This design keeps the public API clean while hiding the complex globalization machinery.

5.5 Proof Architecture 2: The Direct Cuts Path

File: `RepresentationTheorem/Alternative/DirectCuts.lean`

The DirectCuts path provides **both identity-based and identity-free** versions using Dedekind cuts:

- **Identity-free**: Uses `Theta_cuts_pnat` with $\mathbb{N}^+$ iteration
  - `Theta_cuts_pnat` (line 1434): Definition via Dedekind cuts using $\mathbb{N}^+$ iteration
  - `Theta_cuts_pnat_strictMono` (line 1530): Strict monotonicity (fully proven)
  - `Theta_cuts_pnat_add` (line 1636): Additivity (fully proven)
  - No reference to `ident` anywhere
- **Identity-based** (§9a): Uses `Theta_cuts` with $\mathbb{N}$ iteration
  - `iterate_op x 0 = ident` for the base case
  - `ident` as the canonical reference point
  - Produces `RepresentationResult` satisfying $\Theta(\text{ident}) = 0$

The cuts construction uses a classical Hölder/Dedekind approach (shown here for the identity-based version; the identity-free version uses `IsPositive` instead of comparing to `ident`):

1. **Fix base element**: Choose any $a_0 >$ ident as a reference point (identity-free: choose any a_0 with `IsPositive` a_0)

2. **Define rational approximants**: For any $x \in \alpha$, consider the set of ratios $m/n \in \mathbb{Q}$ where $a_0^m \leq x^n$ (equivalently, $m \cdot a_0 \leq n \cdot x$ in additive notation)

3. **Define $\Theta(x)$ by supremum in $\mathbb{R}$**:

$$\Theta_{\text{cuts}}(x) := \sup_{\mathbb{R}}\{m/n \in \mathbb{Q} : a_0^m \leq x^n,\ n > 0\}$$

   where the supremum is taken in $\mathbb{R}$ (which is already complete from Mathlib). The cut set is defined in α using the order relation, but the supremum is computed in $\mathbb{R}$.

4. **Prove properties**:

   - **Order preservation**: If $x < y$, then for any m/n in the cut of x, there exists m'/n' in the cut of y with $m/n < m'/n'$ (uses KSSeparation to find witnesses)
   - **Additivity**: $\Theta(x \oplus y) = \Theta(x) + \Theta(y)$ follows from $a_0^{m_1+m_2} \le (x \oplus y)^{n_1 \cdot n_2}$ iff $a_0^{m_1} \le x^{n_1}$ and $a_0^{m_2} \le y^{n_2}$ (uses commutativity and associativity)

Remark 5.9 (No Circularity). This construction does **not** require completing α into $\mathbb{R}$ first. Instead:

- The set $\{m/n \in \mathbb{Q} : a_0^m \le x^n\}$ is defined using the order relation in α
- These rationals are cast to $\mathbb{R}$: `(↑) '' cutSet` a x `: Set` $\mathbb{R}$
- The supremum is computed in $\mathbb{R}$ using `sSup` (conditional supremum from Mathlib)

Thus $\Theta : \alpha \to \mathbb{R}$ is directly defined without requiring α to already embed into $\mathbb{R}$.

Theorem 5.10 (`associativity_representation_cuts`)**.** ***Lines 43–48, Alternative/Main.lean***
The cuts-based representation theorem.

Listing 13: Appendix A (cuts proof)

```
theorem associativity_representation_cuts
    (alpha : Type*) [KnuthSkillingAlgebra alpha] [KSSeparation alpha]
    [KSSeparationStrict alpha] :
    ∃ Theta : alpha → R,
      (∀ a b : alpha, a ≤ b ↔ Theta a ≤ Theta b) /\
      Theta ident = 0 /\
      ∀ x y : alpha, Theta (op x y) = Theta x + Theta y := by
  -- Use Theta_cuts (the Dedekind-cuts construction)
  obtain <a0, ha0> := <witness for non-trivial element>
  refine <Theta_cuts a0 ha0, order_preservation, identity, additivity>
```

Remark 5.11 (Comparison to Grid Proof). The cuts proof is significantly more compact:

- **Grid proof**: ∼2000+ lines (induction machinery, extension lemmas, path independence)
- **Cuts proof**: ∼500 lines (direct construction, no induction)

However, the grid proof more closely follows K&S's original argument structure (A/B/C partition, δ-choice), while the cuts proof uses the standard Hölder technique from ordered group theory.

Corollary 5.12 (op_comm_of_associativity)**.** ***Lines 65–70, RepresentationTheorem/Main.lean***
Commutativity follows from the representation theorem.

Listing 14: Commutativity from representation

```
theorem op_comm_of_associativity
    (alpha : Type*) [KnuthSkillingAlgebra alpha] [KSSeparation alpha]
    [RepresentationGlobalization alpha] :
    ∀ x y : alpha, op x y = op y x := by
  classical
  obtain <Theta, hTheta_order, _, hTheta_add> := associativity_representation (
      alpha := alpha)
  exact commutativity_from_representation Theta hTheta_order hTheta_add
```

6 The Product Theorem (Appendix B)

Files:

- `ProductTheorem/Main.lean` (K&S's actual path via Appendix A)
- `ProductTheorem/Alternative/DirectProof.lean` (Alternative: direct algebraic path)
- `ProductTheorem/ScaledMultRep.lean` (Common interface for both paths)

6.1 Two Complete Proof Paths

Like Appendix A, the formalization provides **two independent proofs** of Appendix B's conclusion. Both paths arrive at the same result: the tensor operation $\otimes$ on positive reals equals multiplication up to a global scale constant.

6.1.1 Path 1: K&S's Actual Derivation (Recommended)

`ProductTheorem/Main.lean` follows K&S's paper exactly: "apply Appendix A again to $\otimes$". This path uses `AdditiveOrderIsoRep` (from Appendix A) to derive the product equation, then solves it to show $\otimes$ is scaled multiplication.

6.1.2 Path 2: Alternative (Direct Algebraic Proof)

`ProductTheorem/Alternative/DirectProof.lean` provides a direct algebraic proof that any tensor satisfying distributivity (Axiom 3) and associativity (Axiom 4) must be scaled multiplication.

Remark 6.1 (Why Two Paths?).

- **Path 1** assumes existence of `AdditiveOrderIsoRep` for the tensor ("apply Appendix A again")
- **Path 2** derives the same result directly from distributivity + associativity axioms
- Both arrive at the same conclusion: $\otimes$ is scaled multiplication
- **Note**: This is NOT "Aczél's derivation of probability theory" (a separate classical approach); it's just an alternative proof technique for K&S's Appendix B

6.2 Common Interface: ScaledMultRep

Both paths provide the `ScaledMultRep` interface, which captures the OUTPUT of Appendix B:

Listing 15: ScaledMultRep interface (ProductTheorem/ScaledMultRep.lean:44)

```
structure ScaledMultRep (tensor : PosR → PosR → PosR) where
  C : R                 -- The scale constant C > 0
  C_pos : 0 < C
  tensor_eq : ∀ x y : PosR,
    ((tensor x y) : R) = ((x : R) * (y : R)) / C
```

Design principle: Like `AdditiveOrderIsoRep` for Appendix A, this interface captures WHAT Appendix B proves without depending on HOW it was proven. Downstream code (`ConditionalProbability`, `ProbabilityDerivation`, etc.) should depend on `ScaledMultRep`, NOT on specific proof paths.

Constructors:

- `scaledMultRep_of_additiveOrderIsoRep`: K&S path (uses Appendix A)

- `scaledMultRep_of_tensorRegularity`: Direct path (bypasses Appendix A)
- `scaledMultRep_of_assoc_distrib_comm`: Minimal assumptions (assoc + distrib + comm)

6.3 Product-Side Symmetries (3–4)

K&S paper location: Symmetry 3 appears at equation (7) on page 6 (arxiv.tex lines 462–467), Axiom 3 at equation (24) on page 9 (arxiv.tex lines 566–572).

Before applying Appendix A, K&S work with lattice elements and the direct-product operator $\times$. **Symmetry 3** states that $\times$ is (right-)distributive over the join $\sqcup$:

$$(x \times t) \sqcup (y \times t) = (x \sqcup y) \times t$$

After Appendix A provides the representation $\Theta : \alpha \to \mathbb{R}$, we work with graded measures and the tensor operation $\otimes$. **Axiom 3** is the graded version:

$$(x \otimes t) \oplus (y \otimes t) = (x \oplus y) \otimes t$$

After moving to real numbers via Θ, this becomes (Appendix B, arxiv.tex line 661):

$$x \otimes t + y \otimes t = (x + y) \otimes t$$

where $+$ is real addition (since $\oplus$ has been identified with $+$ by Appendix A).

Symmetry 4 (Product Associativity): $(u \otimes v) \otimes w = u \otimes (v \otimes w)$

Formalization note: Our formalization works directly at the graded level (positive reals) rather than formalizing the lattice level. Therefore:

- We define `DistributesOverAdd` as a *property* that a tensor may or may not satisfy
- We then *assume* this property holds for the tensor under consideration
- Ideally, this would be *derived* from Symmetry 3 at the lattice level, but we have not formalized the lattice $\to$ graded transition

Listing 16: Distributivity property (ProductTheorem/Basic.lean:64)

```
-- Defines the PROPERTY (not an axiom, just a predicate)
def DistributesOverAdd (tensor : PosR → PosR → PosR) : Prop :=
  ∀ x y t : PosR, tensor (addPos x y) t = addPos (tensor x t) (tensor y t)

-- Then we ASSUME some tensor satisfies this property:
variable (hDistrib : DistributesOverAdd tensor)
```

Remark 6.2 (Why is this an assumption rather than derived?). In K&S's development:

1. Symmetry 3 is stated at the lattice level: $(x \times t) \sqcup (y \times t) = (x \sqcup y) \times t$
2. After Appendix A provides the representation, this *should* automatically give Axiom 3
3. The transition from lattice to graded measures should preserve this property

Our formalization skips the lattice level and works directly with graded measures (positive reals), so we assume `DistributesOverAdd` as an axiom rather than deriving it from Symmetry 3.

A complete formalization would:

1. Formalize distributive lattices with Symmetry 3
2. Prove Appendix A at the lattice level
3. *Derive* that the graded tensor satisfies `DistributesOverAdd`

Remark 6.3 (Unbundled Tensor Predicates and TensorAlgebra). **Lines 72–115, ProductTheorem/Basic.lean**

Following the unified axiom organization, tensor properties have both unbundled predicates and a bundled class:

Unbundled predicates:

- `TensorAssoc tensor` (line 72): Associativity of $\otimes$
- `TensorPos tensor` (line 77): Positivity-preserving
- `TensorStrictMonoLeft tensor` (line 81): Strict monotonicity in left argument
- `TensorStrictMonoRight tensor` (line 85): Strict monotonicity in right argument
- `DistributesOverAdd tensor` (line 64): Distributivity over +

Bundled class (line 103):

```
class TensorAlgebra (tensor : Posℝ → Posℝ → Posℝ) : Prop where
  distributes : DistributesOverAdd tensor
  assoc : TensorAssoc tensor
  pos : TensorPos tensor
```

Convenience theorem (line 115): `productEquation_of_tensorAlgebra` provides an ergonomic entry point for proofs that use all the bundled axioms together.

Design principle: Use unbundled predicates (e.g., `hDistrib : DistributesOverAdd tensor`) when tracking minimal hypotheses. Use `[TensorAlgebra tensor]` for ergonomic access in longer proofs.

6.4 Product Equation

Appendix B shows $\otimes$ must be multiplication up to a global scale.

Theorem 6.4 (Psi_is_exp). ***Line 43, ProductTheorem/Main.lean***

The inverse representation $\Psi = \Theta^{-1}$ *is exponential:* $\Psi(x) = C \cdot e^{Ax}$ *for some constants* $C > 0$ *and* A*.*

Listing 17: Appendix B: Ψ is exponential

```
theorem Psi_is_exp
    (hRep : AdditiveOrderIsoRep Posℝ tensor)
    (hDistrib : DistributesOverAdd tensor) :
    ∃ (C A : ℝ), 0 < C /\ ∀ x : ℝ, Derived.Psi hRep x = C * ℝ.exp (A * x)
    := by
  refine
    productEquation_solution_of_continuous_strictMono
      (hEq := productEquation_Psi (tensor := tensor) hRep hDistrib)
      (hPos := fun x => Derived.Psi_pos (tensor := tensor) hRep x)
      (hCont := Derived.Psi_continuous (tensor := tensor) hRep)
      (hMono := Derived.Psi_strictMono (tensor := tensor) hRep)
```

Remark 6.5 (The Functional Equation Proof). The proof delegates to:

`productEquation_solution_of_continuous_strictMono` (line 294, `FunctionalEquation.lean`)

This proves a **classical result from functional equations theory**:

Statement: If $\Psi : \mathbb{R} \to \mathbb{R}$ satisfies the product equation

$$\Psi(\tau + \xi) + \Psi(\tau + \eta) = \Psi(\tau + \zeta(\xi, \eta))$$

for all $\tau, \xi, \eta \in \mathbb{R}$, and if Ψ is positive, continuous, and strictly monotone, then $\Psi(x) = C \cdot e^{Ax}$ for some constants $C > 0$ and A.

Key steps (561 lines):

1. Extract shift constant: $a := \zeta(0, 0)$ gives $\Psi(x + a) = 2\Psi(x)$
2. Extend to powers: $\Psi(x + na) = 2^n\Psi(x)$ for all $n \in \mathbb{Z}$
3. Extend to rationals: $\Psi(x + (m/n)a) = 2^{m/n}\Psi(x)$ for all $m/n \in \mathbb{Q}$
4. Use continuity + density: Extend to all reals
5. Conclude: $\Psi(x) = C \cdot 2^{x/a} = C \cdot e^{(\ln 2/a)\cdot x}$

The continuity and monotonicity hypotheses are **derived** (not assumed) from the order isomorphism $\Theta : \mathrm{PosReal} \simeq_o \mathbb{R}$ established in Appendix A:

- `Psi_strictMono` (lines 88–92, `ProductTheorem/Basic.lean`): Since $\Psi := \Theta^{-1}$ and Θ is an order isomorphism, Θ^{-1} is strictly monotone.
- `Psi_continuous` (lines 94–97, `ProductTheorem/Basic.lean`): Order isomorphisms $\mathbb{R} \simeq_o \mathbb{R}$ are continuous (order topology). The proof uses that Θ.symm is continuous, composed with the continuous subtype projection.

Why Exponential Ψ Implies Tensor = Scaled Multiplication

Given: $\Theta(x \otimes y) = \Theta(x) + \Theta(y)$ (additivity) and $\Psi = \Theta^{-1}$ with $\Psi(z) = C \cdot e^{Az}$

Derivation:

$$x = \Psi(\Theta(x)) = C \cdot e^{A\cdot\Theta(x)} \quad \Rightarrow \quad e^{A\cdot\Theta(x)} = x/C$$

$$\begin{aligned} x \otimes y &= \Psi(\Theta(x \otimes y)) = \Psi(\Theta(x) + \Theta(y)) \\ &= C \cdot e^{A(\Theta(x)+\Theta(y))} = C \cdot e^{A\cdot\Theta(x)} \cdot e^{A\cdot\Theta(y)} \\ &= C \cdot (x/C) \cdot (y/C) = \frac{x \cdot y}{C} \end{aligned}$$

Conclusion: $x \otimes y = (x \cdot y)/C$ (Lean: `tensor_coe_eq_mul_div_const`, line 61)

Theorem 6.6 (`tensor_mul_rule_normalized`)**.** ***Line 105, ProductTheorem/Main.lean***
The tensor operation is multiplication up to a global constant: $(x \otimes y)/C = (x/C) \cdot (y/C)$.

Listing 18: Product rule (normalized)

```
theorem tensor_mul_rule_normalized
    (hRep : AdditiveOrderIsoRep PosR tensor)
    (hDistrib : DistributesOverAdd tensor) :
    ∃ C : R, 0 < C /\
      (∀ x : PosR, 0 < ((x : R) / C)) /\
      (∀ x y : PosR,
        ((tensor x y : PosR) : R) / C = (((x : R) / C) * ((y : R) / C)))
```

7 The Variational Theorem (Appendix C)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/VariationalTheorem.lean`

7.1 Variational Functional Equation

What this is about: K&S derive the entropy form $H(m) = A + Bm + C(m \log m - m)$ (the negative of Shannon/Boltzmann entropy) from a variational principle. The key functional equation comes from maximizing H subject to constraints.

The equation: The derivative $H'(m)$ must satisfy:

$$H'(m_x \cdot m_y) = \lambda(m_x) + \mu(m_y)$$

where m_x, m_y are probability masses (positive reals).

Intuition: This says the potential function separates multiplicatively - the derivative at a product decomposes into separate contributions from each factor.

Definition 7.1 (`VariationalEquation`). **Lines 201–202, VariationalTheorem.lean**

Listing 19: Variational equation definition

```
def VariationalEquation (H' lam mu : R → R) : Prop :=
  ∀ m_x m_y : R, 0 < m_x → 0 < m_y → H' (m_x * m_y) = lam m_x + mu m_y
```

7.2 Main Theorem

Result: The only measurable solutions to the variational equation are logarithmic.

Why this matters: This shows that the entropy form is **uniquely determined** (up to constants) by the variational principle plus measurability. You can't have some other weird function satisfy the constraints.

Proof strategy: Transform the multiplicative equation $H'(m_x \cdot m_y) = \lambda(m_x) + \mu(m_y)$ into Cauchy's additive equation $f(u + v) = f(u) + f(v)$ by setting $u = \log m$. Measurable solutions to Cauchy's equation are linear, giving $H'(m) = B + C \log m$.

Theorem 7.2 (`variationalEquation_solution_measurable`). ***Lines 310–375, VariationalTheorem.lean***

If H' satisfies the variational equation and is Borel-measurable, then:

$$H'(m) = B + C \cdot \log(m)$$

for some constants B, C.

Listing 20: Appendix C main theorem

```
theorem variationalEquation_solution_measurable
    (H' : R → R) (lam mu : R → R)
    (hMeas : Measurable H')
    (hV : VariationalEquation H' lam mu) :
    ∃ B C : R, ∀ m : R, 0 < m → H' m = B + C * R.log m := by
  -- Step 1: Extract the common core phi from lam and mu
  obtain <phi, c1, c2, hphi1, hlam, hmu> := hV.∃_common_core
  -- Step 2-7: Transform to Cauchy equation and apply linear solution
  ...
```

7.3 The Entropy Form

What this is: The classical Shannon/Boltzmann entropy appears as the antiderivative of the logarithmic solution.

Derivation: Integrating $H'(m) = B + C\log(m)$ with respect to m:

$$H(m) = \int (B + C\log m)\, dm = A + Bm + C(m\log m - m)$$

where the integration constant is A.

Physical interpretation: The term $m\log m$ is (up to sign and constants) the Shannon entropy $-\sum p_i \log p_i$ for discrete distributions, or the Boltzmann entropy $-\int p(x)\log p(x)\, dx$ for continuous distributions. The other terms (A, Bm) are normalization and constraint adjustments.

Why it matters: This shows that entropy **isn't an axiom** - it's derived from the variational principle applied to the K&S probability framework.

Definition 7.3 (`entropyForm`)**. Line 475, VariationalTheorem.lean**

Listing 21: Entropy form

```
noncomputable def entropyForm (A B C : ℝ) : ℝ → ℝ :=
  fun m => A + B * m + C * (m * ℝ.log m - m)
```

Definition 7.4 (`entropyDerivative`)**. Line 287, VariationalTheorem.lean**

The expected derivative of the entropy form.

Listing 22: Expected derivative: $B + C\log m$

```
noncomputable def entropyDerivative (B C : ℝ) : ℝ → ℝ :=
  fun m => B + C * ℝ.log m
```

Theorem 7.5 (`entropyForm_deriv`)**. *Lines 478–494, VariationalTheorem.lean***

The entropy form has derivative $H'(m) = B + C\log m$.

Listing 23: Proof that entropy form has the correct derivative

```
theorem entropyForm_deriv (A B C : ℝ) {m : ℝ} (hm : 0 < m) :
    HasDerivAt (entropyForm A B C) (entropyDerivative B C m) m := by
  unfold entropyForm entropyDerivative
  -- d/dm [A + Bm + C(m log m - m)] = B + C(log m + 1 - 1) = B + C log m
  ...
```

Remark 7.6 (What this proves). The theorem `entropyForm_deriv` proves that:

$$\frac{d}{dm}\left[A + Bm + C(m\log m - m)\right] = B + C\log m$$

This verifies that integrating the logarithmic solution $H'(m) = B + C\log m$ (from the variational equation) gives the entropy form $H(m) = A + Bm + C(m\log m - m)$.

8 Divergence (K&S Section 6)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/Divergence.lean`

What this is: The divergence $\phi(w, u)$ measures the "distance" between two measure assignments w and u. It quantifies the information-theoretic cost of using measure u when the "true" measure is w.

Connection to Appendix C (Entropy Form): The divergence is a **special case** of the entropy form from the variational theorem:

$$H(m) = A + Bm + C(m \log m - m) \quad \text{(Appendix C)}$$
$$\phi(w, u) = u - w + w \log(w/u) \quad \text{(Divergence)}$$

Setting $A = u$, $B = -\log(u)$, $C = 1$ in the entropy form gives the divergence (K&S Eq. 44). The critical point analysis from Appendix C proves that $\phi(w, u)$ is minimized when $w = u$.

Note: This is **atom divergence** for general real-valued measures. The specialization to **probability distributions** happens in Section 10, after Section 9 derives what probability distributions are.

Key properties:

- **Non-negative**: $\phi(w, u) \geq 0$, with equality iff $w = u$ (formalized in `atomDivergence_nonneg`, lines 102–120)
- **Asymmetric**: $\phi(w, u) \neq \phi(u, w)$ in general (it's NOT a distance metric)
- **Connects variational calculus to information theory**: Bridge between Appendix C and Section 8

Forward reference: This atom divergence will be specialized to **probability distributions** in Section 10 (Information and Entropy), giving the Kullback-Leibler divergence formula.

Definition 8.1 (`atomDivergence`)**. Lines 68–69, Divergence.lean**
The per-atom divergence: $\phi(w, u) = u - w + w \log(w/u)$.

Listing 24: Atom divergence

```
noncomputable def atomDivergence (w u : R) : R :=
  u - w + w * log (w / u)
```

Theorem 8.2 (`atomDivergence_nonneg`)**. *Lines 102–120, Divergence.lean***

Listing 25: Divergence non-negativity

```
theorem atomDivergence_nonneg (w u : R) (hw : 0 < w) (hu : 0 < u) :
    0 ≤ atomDivergence w u := by
  unfold atomDivergence
  -- Rewrite as w * (u/w - 1 - log(u/w)) and use log inequality
  let s := u / w
  have hs : 0 < s := div_pos hu hw
  have hrewrite : u - w + w * log (w / u) = w * (s - 1 - log s) := by ...
  rw [hrewrite]
  exact mul_nonneg (le_of_lt hw) (log_ineq s hs)
```

Theorem 8.3 (`atomDivergence_eq_zero_iff`)**.** ***Lines 123–159, Divergence.lean***
Divergence equals zero if and only if $w = u$.

Listing 26: Divergence equals zero iff $w = u$ (Divergence.lean:123)

```
theorem atomDivergence_eq_zero_iff (w u : R) (hw : 0 < w) (hu : 0 < u) :
    atomDivergence w u = 0 ↔ w = u := by
  constructor
  . -- If phi(w,u) = 0, then w = u
    intro h
    let s := u / w
    have hs : 0 < s := div_pos hu hw
    -- phi(w,u) = w * (s - 1 - log s) = 0
    -- w > 0 implies s - 1 - log s = 0
    -- But s - 1 - log s > 0 for s ≠ 1 (strict log inequality)
    -- So s = 1, hence u = w
    ...
  .
    intro heq
    rw [heq]
    exact atomDivergence_self u hu
```

9 Conditional Probability (K&S Section 7)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/ConditionalProbability/Basic.lean`

What this section does: K&S Section 7 derives **probability calculus** from first principles. This is the crucial step that takes us from general measures (Sections 1–6) to **probability distributions** (normalized measures).

Starting with conditional plausibility as a **bivaluation** $p(x|t)$ (a function taking pairs of lattice elements to reals), K&S introduces **Axiom 5 (Chaining Associativity)** and proves:

1. The **chain-product rule**: $\Pr(a|c) = \Pr(a|b) \cdot \Pr(b|c)$ for chains $a \le b \le c$
2. **Bayes' theorem**: $\Pr(x|\theta) \cdot \Pr(\theta) = \Pr(\theta|x) \cdot \Pr(x)$
3. **Probability as a ratio**: $\Pr(x|t) = \frac{m(x \wedge t)}{m(t)}$ (K&S Eq. 53)

Key insight: The SAME functional equation from Appendix B (product equation) reappears here! Axiom 5 + sum rule forces the chaining operation to be multiplication (up to scale).

Deliverable: This section establishes that **probability is a normalized measure** - "simply the shape of the confined measure, automatically normalized to unit mass" (K&S, Section 7.3). This gives us **probability distributions**, which are used in Section 10.

9.1 Structural Change: From Linear Order to Lattice

Remark 9.1 (Different Type Structure)*.* K&S Section 7 operates on a **different type** than Sections 1–6:

- **Sections 1–6** (K&S algebra): `[LinearOrder` α`]` - measures on linearly ordered values
- **Section 7** (Bivaluation): `[Lattice` α`] [BoundedOrder` α`]` - probability on lattice of events

This reflects K&S's conceptual shift: earlier sections study **measure values** (which are linearly ordered reals), while Section 7 studies **conditional probability on events** (which form a lattice).

The Lean formalization respects the **logical dependency order**, not K&S's presentation order. Section 7 requires lattice operations ($\wedge$, $\vee$, $\perp$, $\top$) that weren't needed in Sections 1–6.

Lattice hierarchy in the code:

- `Bivaluation` structure (line 59): `[Lattice α]` - general lattice
- Main theorems (chain-product, Bayes): `[DistribLattice α]` - needs distributivity
- Optional theorems (`sumRule_general`, `complementRule`): `[BooleanAlgebra α]` - needs complements

Mathematical generalization: K&S works with propositions (Boolean), but the Lean formalization proves the core probability calculus works on **any distributive lattice**. Boolean structure is only needed for complement operations.

9.2 Bivaluation and Axiom 5

Definition 9.2 (`Bivaluation`)**. Lines 59–73, Basic.lean**

A bivaluation $p : \alpha \to \alpha \to \mathbb{R}$ represents conditional plausibility on a lattice with:

- **Positivity**: $p(x|t) > 0$ when $\perp < x \leq t$
- **Sum rule**: $p(x \vee y|t) = p(x|t) + p(y|t)$ for disjoint x, y
- **Context intersection**: $p(x|t) = p(x \wedge t|t)$ (implicit in K&S)

Remark 9.3 (Lattice Structure on Events, Not Context)*.* Note that the sum rule applies to the **first argument** (the event), not the context:

$$p(x \vee y \mid t) = p(x \mid t) + p(y \mid t)$$

The context t stays **fixed** while events are decomposed via the lattice join $\vee$. This matches the standard probability identity $P(A \cup B \mid C) = P(A \mid C) + P(B \mid C)$ for disjoint A, B. The lattice operations ($\vee$, $\wedge$, $\perp$) describe the *event algebra*; the context is just a parameter.

Definition 9.4 (Axiom 5: Chaining Associativity)**. Lines 109–128 (ChainingOp structure), Basic.lean**

The chaining operation $\odot : \mathbb{R} \to \mathbb{R} \to \mathbb{R}$ on plausibility values satisfies:

Listing 27: `ChainingOp` structure (Basic.lean:124)

```
structure ChainingOp where
  chain : R → R → R
  chain_assoc : ∀ x y z, chain (chain x y) z = chain x (chain y z)
  chain_strictMono_left : ∀ z, 0 < z → StrictMono (fun x => chain x z)
  chain_strictMono_right : ∀ x, 0 < x → StrictMono (fun z => chain x z)
  chain_pos : ∀ x y, 0 < x → 0 < y → 0 < chain x y
  chain_distrib_left : ∀ a b t, 0 < a → 0 < b → 0 < t →
    chain a t + chain b t = chain (a + b) t
```

Formulation: For a chain $a < b < c < d$:

$$(p(a|b) \odot p(b|c)) \odot p(c|d) = p(a|b) \odot (p(b|c) \odot p(c|d))$$

Definition 9.5 (Chain Rule)**. Line 219, Basic.lean**

The chain rule connects the chaining operation to the bivaluation:

Listing 28: `ChainingAssociativity` class (Basic.lean:219)

```
class ChainingAssociativity (alpha : Type*) [Lattice alpha] [BoundedOrder alpha]
    (B : Bivaluation alpha) where
  chainOp : ChainingOp
  chain_rule : ∀ a b c : alpha, a ≤ b → b ≤ c → Bot < a →
    B.p a c = chainOp.chain (B.p a b) (B.p b c)
```

This says: $p(a|c) = p(a|b) \odot p(b|c)$ for chains $a \leq b \leq c$.[1]

9.3 The Product Equation Reappears

Lines 245–314, Basic.lean

K&S's brilliant observation: combining chaining associativity with the sum rule gives the **exact same product equation** as Appendix B!

Proof sketch:

1. Let Θ be the function such that $\Theta(p(a|b) \odot p(b|c)) = \Theta(p(a|b)) + \Theta(p(b|c))$
2. Define $\Psi = \Theta^{-1}$
3. The sum rule says: for disjoint x, y with intermediate context,
$$\mathrm{chain}(a, t) + \mathrm{chain}(b, t) = \mathrm{chain}(a + b, t)$$
This is **left-distributivity over addition** - exactly the Appendix B hypothesis!
4. Therefore: $\Psi(\tau + \xi) + \Psi(\tau + \eta) = \Psi(\tau + \zeta(\xi, \eta))$ where $\zeta(\xi, \eta) = \Theta(\Psi(\xi) + \Psi(\eta))$
5. By Appendix B: $\Theta = A \cdot \log$ for some $A > 0$
6. Hence the chaining operation is: $\mathrm{chain}(x, y) = \frac{x \cdot y}{K}$ for some $K > 0$

9.4 Chain-Product Rule

Theorem 9.6 (`chainProductRule`)**.** ***Line 345, Basic.lean***

For chains $a \leq b \leq c$ in a lattice with normalized bivaluation ($p(t|t) = 1$):

$$\Pr(a|c) = \Pr(a|b) \cdot \Pr(b|c)$$

Listing 29: `chainProductRule` (Basic.lean:345)

```
theorem chainProductRule
    {alpha : Type*} [DistribLattice alpha] [BoundedOrder alpha]
    (B : Bivaluation alpha) [CA : ChainingAssociativity alpha B]
    (hNormalized : ∀ t : alpha, Bot < t → B.p t t = 1) :
    ∀ a b c : alpha, a ≤ b → b ≤ c → Bot < a →
      B.p a c = B.p a b * B.p b c
```

[1] K&S uses "interval notation" $[x, y]$ throughout Section 7 without formally defining intervals as mathematical objects. They write $\alpha = [x, y]$, $\beta = [y, z]$, etc., and speak of "concatenating intervals" $[x, y] \circ [y, z] = [x, z]$. The chaining operation $\odot$ then acts on the plausibility *values* of these intervals: $p(\alpha) \odot p(\beta)$. Our formalization sidesteps this implicit interval semantics by working directly with lattice elements: the "interval $[a, b]$" is represented implicitly by the pair (a, b) with constraint $a \leq b$. The chain rule then states $p(a|c) = p(a|b) \odot p(b|c)$ for $a \leq b \leq c$, which captures the compositional structure without reifying intervals as first-class objects.

Proof strategy:

- Appendix B gives: $\text{chain}(x, y) = (x \cdot y)/K$ for some $K > 0$
- Normalization at (a, a, a) forces $K = 1$: since $p(a|a) = 1$, we have $1 = \text{chain}(1, 1) = 1/K$
- Therefore: $p(a|c) = \text{chain}(p(a|b), p(b|c)) = p(a|b) \cdot p(b|c)$

9.5 Bayes' Theorem

Theorem 9.7 (`bayesTheorem`)**. *Line 421, Basic.lean***
For $x, \theta \leq t$ in a distributive lattice:

$$\Pr(x|\theta) \cdot \Pr(\theta|t) = \Pr(\theta|x) \cdot \Pr(x|t)$$

Listing 30: `bayesTheorem` (Basic.lean:421)

```
theorem bayesTheorem
    {alpha : Type*} [DistribLattice alpha] [BoundedOrder alpha]
    (B : Bivaluation alpha) [CA : ChainingAssociativity alpha B]
    (hNormalized : ∀ t : alpha, Bot < t → B.p t t = 1)
    (x theta t : alpha) (hxtheta_pos : Bot < x ⊓ theta) (hx : x ≤ t) (htheta :
        theta ≤ t)
    (hx_pos : Bot < x) (htheta_pos : Bot < theta) :
    B.p x theta * B.p theta t = B.p theta x * B.p x t
```

Proof: Both sides equal $\Pr(x \wedge \theta|t)$ by the product rule and commutativity of $\wedge$.

9.6 Probability as Ratio of Measures

Theorem 9.8 (`prob_eq_measure_ratio`)**. *Line 462, Basic.lean***
*Define the **unconditional measure** by $m(x) := p(x|\top)$. Then for any context $t \neq \bot$:*

$$\Pr(x|t) = \frac{m(x \wedge t)}{m(t)}$$

Listing 31: `prob_eq_measure_ratio` (Basic.lean:462)

```
theorem prob_eq_measure_ratio
    {alpha : Type*} [DistribLattice alpha] [BoundedOrder alpha]
    (B : Bivaluation alpha) [CA : ChainingAssociativity alpha B]
    (hNormalized : ∀ t : alpha, Bot < t → B.p t t = 1) :
    ∀ x t : alpha, t ≠ Bot → B.p x t = baseMeasure B (x ⊓ t) / baseMeasure B t
```

This single formula subsumes the sum rule, chain-product rule, and range $[0, 1]$. Probability is simply the **ratio of measures** — "the elementary calculus of proportions of measure" (K&S, Section 7.3).

9.7 baseMeasure Satisfies Measure Axioms

Lines 559–576, Basic.lean
The derived `baseMeasure` satisfies the classical measure axioms:

Theorem 9.9 (`baseMeasure_satisfies_measure_axioms`)**.** *For a normalized Bivaluation ($p(t|t) = 1$ for $t > \bot$), `baseMeasure` is a probability measure:*

1. $m(\perp) = 0$ *(empty set has measure zero)*

2. *Finite additivity:* $m(x \vee y) = m(x) + m(y)$ *for disjoint* x, y

3. *Non-negativity:* $0 \leq m(x)$

4. *Normalization:* $m(\top) = 1$

Listing 32: `baseMeasure_satisfies_measure_axioms` (Basic.lean:559)

```
theorem baseMeasure_satisfies_measure_axioms
    (hNormalized : ∀ t : alpha, Bot < t → B.p t t = 1)
    (hTop : (Top : alpha) ≠ Bot) :
    baseMeasure B Bot = 0 /\
    (∀ x y : alpha, Disjoint x y →
      baseMeasure B (x ⊔ y) = baseMeasure B x + baseMeasure B y) /\
    (∀ x : alpha, 0 ≤ baseMeasure B x) /\
    baseMeasure B Top = 1
```

Key point: For finite Boolean algebras, finite additivity is equivalent to σ-additivity, so this is a bona fide probability measure in the Kolmogorov sense.

Remark 9.10 (Additional Measure Properties). The formalization also proves:

- **Inclusion-exclusion** (`baseMeasure_inclusion_exclusion`, line 588):

$$m(x \vee y) + m(x \wedge y) = m(x) + m(y)$$

- **Complement rule** (`baseMeasure_compl_normalized`, line 640):

$$m(x^c) = 1 - m(x)$$

- **Subadditivity** (`baseMeasure_subadditive`, line 649):

$$m(x \vee y) \leq m(x) + m(y)$$

- **ENNReal version** (`baseMeasureENNReal`, line 673): For Mathlib compatibility

10 Information and Entropy (K&S Section 8)

File: `Mettapedia/ProbabilityTheory/KnuthSkilling/InformationEntropy.lean`

What this section does: K&S Section 8 takes **special cases** of the variational potential H from Appendix C, specialized to probability distributions (normalized measures from Section 7).

Key point: Shannon entropy is **derived**, not just defined. It emerges as an "inevitable consequence of seeking a variational quantity" (K&S, Section 8.2).

10.1 From Atom Divergence to KL Divergence

The key step: Now that we have probability distributions from Section 9, we can specialize the atom divergence from Section 8 to normalized measures.

For probability distributions $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ where $\sum p_i = 1$ and $\sum q_i = 1$:

$$\begin{aligned}
\sum_i \phi(p_i, q_i) &= \sum_i (q_i - p_i + p_i \log(p_i/q_i)) \\
&= \underbrace{\sum_i q_i}_{=1} - \underbrace{\sum_i p_i}_{=1} + \sum_i p_i \log(p_i/q_i) \\
&= \sum_i p_i \log(p_i/q_i) = D_{KL}(P \| Q)
\end{aligned}$$

This is the **Kullback-Leibler divergence** (K&S Eq. 54).

Formalized in Lean (`klDivergence_from_divergence_formula`, lines 211–224, `InformationEntropy.lean`):

```
theorem klDivergence_from_divergence_formula (P Q : ProbDist n)
    (hQ_pos : ∀ i, P.p i ≠ 0 → 0 < Q.p i) :
    klDivergence P Q hQ_pos =
      sum i, atomDivergence (P.p i) (Q.p i) - (sum i, Q.p i - sum i, P.p i)
```

The proof uses the normalization constraints: $\sum(q_i - p_i) = 1 - 1 = 0$, so the linear terms cancel.

10.2 Derivation Chain

From Appendix C to Shannon Entropy:

1. **Appendix C** establishes the general variational form for any measure:

$$H(m) = A + B \cdot m + C \cdot (m \log m - m)$$

2. **Section 8** specializes to atom divergence: $\phi(w, u) = u - w + w \log(w/u)$

3. **Section 9** proves that probability is a normalized measure: $\Pr(x|t) = m(x \wedge t)/m(t)$

4. **Section 10 (above)**: For probability distributions, divergence simplifies to KL divergence

5. **Section 8.2 (Entropy)**: To quantify uncertainty, we require:

   - Zero uncertainty when one $p_k = 1$ (fully determined state)
   - This forces: $A_k = 0$ and $B_k = C$
   - Setting $C = -1$ (conventional scale) gives:

$$S(p) = -\sum_k p_k \log p_k$$

This is Shannon entropy - not assumed, but **derived from the variational principle**.

10.3 Shannon's Three Properties

K&S claim these properties are "inevitable consequences" (K&S, Section 8.2):

1. **Continuity**: S is a continuous function of its arguments
2. **Monotonicity**: If there are n equal choices ($p_k = 1/n$), then S increases in n
3. **Grouping**: If a choice is broken down into subchoices, S adds according to expectation:

$$S(p_1, p_2, p_3) = S(p_1, p_2 + p_3) + (p_2 + p_3) \cdot S\left(\frac{p_2}{p_2 + p_3}, \frac{p_3}{p_2 + p_3}\right)$$

These are Shannon's original axioms (Shannon 1948). K&S shows they follow from the variational framework.

10.4 Formalization

Definition 10.1 (`ProbDist`)**. Lines 94–98, InformationEntropy.lean**
A probability distribution: probabilities for n outcomes that are non-negative and sum to 1.

Listing 33: Probability distribution

```
structure ProbDist (n : N) where
  p : Fin n → R
  nonneg : ∀ i, 0 ≤ p i
  sum_one : sum i, p i = 1
```

Definition 10.2 (`klDivergence`)**. Line 184, InformationEntropy.lean**
The Kullback-Leibler divergence for probability distributions (K&S Eq. 54).

Listing 34: `klDivergence` (InformationEntropy.lean:184)

```
noncomputable def klDivergence {n : N} (P Q : ProbDist n)
    (hQ_pos : ∀ i, P.p i ≠ 0 → 0 < Q.p i) : R :=
  sum i, P.p i * log (P.p i / Q.p i)
```

Note: The positivity hypothesis `hQ_pos` ensures Q is strictly positive on the support of P, avoiding $\log(p/0)$ issues. This is the regime where K&S's formula is meaningful. An extended version `klDivergenceTop` (line 197) takes values in $\mathbb{R}_{\geq 0} \cup \{\infty\}$, returning ∞ when this condition fails.

The Shannon entropy is formalized as:

$$S(p) = -\sum_i p_i \log(p_i)$$

with the convention $0 \cdot \log(0) = 0$ (from continuity: $\lim_{x \to 0^+} x \log x = 0$), justified by `zero_mul_log_zero` (line 70).

11 Counterexamples and Clarifications

Directory: `Mettapedia/ProbabilityTheory/KnuthSkilling/Counterexamples/`

11.1 The "Discontinuous Re-grading" Claim

K&S (Section 2) claim that continuity is "merely a convenient convention" and suggest a discontinuous "re-grading" map Θ could preserve the sum rule, using a base-conversion example.

Theorem 11.1 (regrade_preserving_sum_rule_is_continuous)**.** ***File:*** *Counterexamples/RegradeCounterexa*
This claim is false. *Any re-grading $\Theta : \mathbb{R} \to \mathbb{R}$ that preserves:*

1. *The sum rule: $\Theta(x + y) = \Theta(x) + \Theta(y)$ (additivity)*
2. *Monotonicity: $x \leq y \Rightarrow \Theta(x) \leq \Theta(y)$*

must be linear ($\Theta(x) = c \cdot x$ for some constant c), hence continuous.

Listing 35: Monotone additive functions are linear (RegradeCounterexample.lean:100)

```
theorem monotone_additive_is_linear {f : R → R}
    (hadd : ∀ x y, f (x + y) = f x + f y)
    (hmono : Monotone f) :
    ∀ x, f x = f 1 * x

theorem regrade_preserving_sum_rule_is_continuous {Theta : R → R}
    (hTheta_add : ∀ x y, Theta (x + y) = Theta x + Theta y)
    (hTheta_mono : Monotone Theta) :
    Continuous Theta
```

Remark 11.2 (Why K&S's Example Fails)*.* K&S's base-conversion map is not additive: $\Theta(x+y) \neq \Theta(x)+\Theta(y)$. Their example could only work by changing the addition operation to some weird $\oplus$, which is just obfuscating notation, not demonstrating genuine discontinuity.

The philosophical point (finite systems can't detect continuity) may be valid, but the mathematical example does not support the claim.

11.2 Pathological Additive Functions

File: `Counterexamples/CauchyPathology.lean`

Without regularity conditions, Cauchy's equation $f(x+y) = f(x)+f(y)$ has "wild" non-linear solutions (constructed via Hamel bases over $\mathbb{Q}$). These solutions are necessarily **non-monotonic**—they oscillate wildly and cannot preserve order.

11.2.1 The Construction

Step 1: Build a Hamel basis (lines 35–77):

We work with $\mathbb{R}$ as a vector space over $\mathbb{Q}$. First prove $\{1, \sqrt{2}\}$ is $\mathbb{Q}$-linearly independent (using irrationality of $\sqrt{2}$), then extend to a Hamel basis:

Listing 36: Hamel basis extending $\{1, \sqrt{2}\}$

```
theorem linearIndepOn_one_sqrt2 :
    LinearIndepOn Q id ({(1 : R), R.sqrt 2} : Set R)

noncomputable def hamelBasis :
    Module.Basis (...extend {1, sqrt 2}...) Q R :=
  Module.Basis.extend linearIndepOn_one_sqrt2
```

Step 2: Define the weird map (lines 79–86):

Create a $\mathbb{Q}$-linear map that sends:

- $1 \mapsto 0$
- $\sqrt{2} \mapsto 1$
- All other basis vectors $\mapsto 0$

Listing 37: Definition of weirdAdditive

```
noncomputable def weirdQLinear : R →_[Q] R :=
  (hamelBasis).constr Q fun i =>
    if (i : R) = R.sqrt 2 then (1 : R) else 0

noncomputable def weirdAdditive : R → R := fun x => weirdQLinear x
```

Step 3: Prove it's additive but not linear (lines 87–127):

Listing 38: weirdAdditive satisfies Cauchy's equation but isn't linear

```
theorem weirdAdditive_add (x y : R) :
    weirdAdditive (x + y) = weirdAdditive x + weirdAdditive y

theorem weirdAdditive_not_mul (A : R) :
    ∃ x : R, weirdAdditive x ≠ A * x
  -- Proof: weirdAdditive 1 = 0 but weirdAdditive (sqrt 2) = 1
  -- So it can't be x |→ A*x for any constant A
```

Step 4: Convert to positive reals (lines 129–165):
Define $H'(m) :=$ weirdAdditive$(\log m)$ on positive reals:

Listing 39: Multiplicative-additive pathology on positive reals

```
noncomputable def Hprime (m : R) : R := weirdAdditive (R.log m)

theorem Hprime_mul (m_x m_y : R) (hx : 0 < m_x) (hy : 0 < m_y) :
    Hprime (m_x * m_y) = Hprime m_x + Hprime m_y

theorem Hprime_not_B_add_C_log :
    ~ ∃ (B C : R), ∀ m, 0 < m → Hprime m = B + C * log m
  -- Proof: If Hprime m = B + C*log m, then weirdAdditive x = C*x,
  -- contradicting weirdAdditive_not_mul
```

Key insight: These pathological solutions exist but **cannot be monotone**. By the theorem in §11.1, any monotone additive function is linear, so wild solutions like `weirdAdditive` must oscillate wildly and violate order preservation.

12 Summary: Complete Formalization

12.1 Major Formalized Theorems at a Glance

This section consolidates all the main results proven in the formalization. All paths are relative to `Mettapedia/ProbabilityTheory/KnuthSkilling/`.

Sum Rule (Appendix A)

- `representation_semigroup` (`Separation/HolderEmbedding.lean:297`) — $\exists\Theta$ preserving order and addition (identity-free)

- `representation_from_noAnomalousPairs` (`Separation/HolderEmbedding.lean:269`) — Full representation with $\Theta(\text{ident}) = 0$
- `associativity_representation` (`RepresentationTheorem/Main.lean:34`) — Public API for Appendix A
- `op_archimedean_of_separation` (`Separation/SandwichSeparation.lean:166`) — Archimedean is derivable

Product Rule (Appendix B)

- `psi_is_exponential` (`ProductTheorem/ProductExp.lean:156`) — $\Psi(x \otimes y) = \Psi(x) \cdot \Psi(y)$
- `product_rule_normalized` (`ProductTheorem/Main.lean:198`) — $p(xy|I) = p(x|I) \cdot p(y|xI)$
- `ScaledMultRep` (`ProductTheorem/ScaledMultRep.lean:44`) — Common interface for both proof paths

Variational (Appendix C)

- `variational_implies_entropy_form` (`VariationalTheorem.lean:167`) — Measurable solution $\Rightarrow H(m) = B + C \log m$
- `entropy_form_satisfies_variational` (`VariationalTheorem.lean:201`) — Converse: entropy form satisfies the equation
- `entropy_form_deriv_correct` (`VariationalTheorem.lean:234`) — Derivative is $B+C \log m$

Conditional Probability (Section 7)

- `chainProductRule` (`ConditionalProbability/Basic.lean:348`) — $p(xy|z) = p(x|z) \cdot p(y|xz)$
- `bayesTheorem` (`ConditionalProbability/Basic.lean:424`) — $p(x|yz) \cdot p(y|z) = p(y|xz) \cdot p(x|z)$
- `prob_eq_measure_ratio` (`ConditionalProbability/Basic.lean:714`) — $p(x|y) = m(x \wedge y)/m(y)$
- `baseMeasure_satisfies_measure_axioms` (`ConditionalProbability/Basic.lean:559`) — m is a probability measure

Divergence & Entropy (Sections 6, 8)

- `atomDivergence_nonneg` (`Divergence.lean:89`) — $D(w\|u) \geq 0$
- `atomDivergence_eq_zero_iff` (`Divergence.lean:123`) — $D(w\|u) = 0 \Leftrightarrow w = u$
- `klDivergence` (`InformationEntropy.lean:184`) — KL divergence from atom divergence
- `shannonEntropy` (`InformationEntropy.lean:267`) — $H = -\sum p \log p$

Quantum Theory Classification (Section 4)

- `selection_theorem` (`SymmetricalFoundation.lean:312`) — $\mu < 0$ gives QM Born rule
- `mean_bornRule_sum_unit_phases` (`SymmetricalFoundation.lean:401`) — Born rule from averaging
- `classification_theorem` (`TwoDimClassification.lean:178`) — Exactly 3 algebra classes

Counterexamples & Clarifications

- `monotone_additive_is_linear` (`Counterexamples/RegradeCounterexample.lean:100`) — Discontinuous re-grading impossible
- `weirdAdditive_add` (`Counterexamples/CauchyPathology.lean:87`) — Pathological additive functions exist
- `Hprime_not_B_add_C_log` (`Counterexamples/CauchyPathology.lean:156`) — But they violate regularity

12.2 What K&S Claims vs. What We Prove

K&S Claim	Lean Status	Notes
Axioms 0–2 $\Rightarrow$ Sum Rule	Proven	Appendix A representation
Archimedean derivable	Proven	From KSSeparation
Commutativity derivable	Proven	From KSSeparation
Axioms 3–4 $\Rightarrow$ Product Rule	Proven	Appendix B
Variational $\Rightarrow$ Entropy form	Proven	Appendix C
3 algebra classes	Proven	TwoDimClassification
$\mu < 0$ for QM	Proven	selection_theorem
Born rule from averaging	Proven	mean_bornRule_sum_unit_phases

12.3 Key Discoveries from Formalization

1. **Linear order is implicit**: K&S proofs assume trichotomy without stating it.
2. **Identity element is proven optional**: K&S say the bottom element is "optional" (lines 320, 340–341). Our formalization **proves** this rigorously via the `KSSemigroupBase` hierarchy:

   Identity-free representation (`HolderEmbedding.lean`):

   - `representation_semigroup` (lines 297–303) proves the representation theorem **without identity**: there exists $\Theta : \alpha \to \mathbb{R}$ with order-preservation and additivity
   - Uses Eric Luap's `OrderedSemigroups` library via `holder_embedding_of_noAnomalousPairs`
   - The Hölder/Alimov embedding theorem works on any cancellative ordered semigroup without anomalous pairs

   What identity provides (`HolderEmbedding.lean:310--313`):

   - `identity_gives_canonical_normalization`: Identity gives $\Theta(\text{ident}) = 0$
   - Without identity, Θ is defined only up to an additive constant c: if Θ is valid, so is $\Theta + c$

- Identity pins down $c = 0$ by requiring $\Theta(\text{ident}) = 0$

Two Appendix A paths:

- **Hölder path** (`HolderEmbedding.lean`): Identity-FREE—produces `RepresentationResult`
- **Grid/Globalization path** (`RepresentationTheorem/`): Currently uses identity—produces `NormalizedRepresentationResult` with $\Theta(\text{ident}) = 0$
- **DirectCuts path** (`Alternative/DirectCuts.lean`): Explicitly requires identity for the cut construction

Grid path identity-free infrastructure (exists but not yet instantiated):

- `AtomFamily_param`: Uses `IsPositive (atoms i)` instead of `ident < atoms i`
- `mu_param`, `kGrid_param`: Parametric versions with explicit base element
- `mu_pnat`, `kGrid_pnat`: Truly identity-free using $\mathbb{N}^+$ iteration
- `RepresentationGlobalizationAnchor`: Class for identity-free representations (normalizes to arbitrary anchor)

See `MultiGrid.lean` section "Identity-Free Grid Definitions" for the parametric API.

Most of the formalization uses `KnuthSkillingAlgebraBase` (with identity) for historical reasons—the development was done with identity first—(4000+ references), but the core representation theorem is proven at the identity-free `KSSemigroupBase` level. K&S are **correct** that identity is optional; we have **proven this rigorously**.

3. **Separation property is necessary**: The representation theorem requires an explicit "separation" axiom to enable rational approximation.

4. **Archimedean is derivable**: Not an axiom—follows from separation.

5. **Commutativity is derivable**: Not an axiom—follows from separation via mass counting.

6. **Classification gives isomorphism**: The original K&S classification theorem is about *isomorphism classes*, not equality of multiplication rules.

7. **Measurability replaces continuity**: For Appendix C, measurability (not differentiability) is the correct regularity assumption.

8. **Discontinuous re-grading is impossible**: K&S's claim that continuity is optional is false for maps preserving both the sum rule and monotonicity (proven in `RegradeCounterexample.lean`).

9. **Symmetries 3–4 are product-side**: The product-side symmetries (distributivity, product associativity) are logically separate from the sum-side axioms and are formalized in `ProductTheorem/Main.lean`.

10. **Interface design pattern**: Both Appendix A and Appendix B use an **interface + multiple implementations** pattern:

    - **Appendix A**: `AdditiveOrderIsoRep` interface with Hölder and Grid implementations
    - **Appendix B**: `ScaledMultRep` interface with K&S path and Direct path implementations

    Downstream code depends only on the interfaces, not on specific proof paths. This separation of concerns allows switching implementations without changing dependent code.

12.4 Sorry Count

Total sorries in core K&S files: 0

All core theorems are fully proven. The formalization is complete and ready for review.

13 Build Instructions

From the Mettapedia project root:

```
export LAKE_JOBS=3
nice -n 19 lake build Mettapedia.ProbabilityTheory.KnuthSkilling
```

For memory-intensive files (ThetaPrime.lean):

```
ulimit -Sv 6291456
export LAKE_JOBS=1
nice -n 19 lake build Mettapedia.ProbabilityTheory.KnuthSkilling.
    RepresentationTheorem
```