Master Thesis Proposal

AI-Assisted Formalization of Bregman Geometry and Mirror Learning in Lean 4

Duration: 6 months **Type:** library-building thesis with a fully verified core theorem

Project idea. Mirror descent, FTRL, Bregman divergences, variational inequalities, and monotone operators are central tools in online optimization and learning in games. Informal proofs often look short, but they are assumption-sensitive: convexity, differentiability, minimizer optimality, dual pairings, step sizes, and telescoping identities must all match exactly. This thesis builds a reusable Lean 4 library for these Bregman and variational foundations.

The project is explicitly AI-assisted. AI tools may be used for lemma search, tactic suggestions, refactoring, theorem-shape exploration, and documentation. The final mathematical correctness must come from Lean 4: all central statements should compile without hidden axioms.

Central mathematical target. The main verified theorem is a mirror-descent theorem built from the Bregman three-point identity:

> Formalize Bregman divergence and an abstract mirror step satisfying the standard optimality inequality, then prove a finite-horizon mirror-descent regret or Lyapunov bound by telescoping Bregman divergences.

The thesis should be ambitious as a library project while keeping the theorem statement precise enough to finish in six months. The preferred setting is finite-dimensional real vector spaces or a robust real normed-space setting already supported by `mathlib`. General Banach-space stochastic approximation, weak compactness, measure-valued strategies, and symplectic dynamics are future extensions, not part of the core target.

Mathematical scope. The library should formalize convex feasible sets, pseudo-gradients, variational inequality solutions, monotone operators, Bregman divergence, three-point identities, mirror steps, and finite-horizon regret/descent inequalities. A convex-game equilibrium is represented as a variational inequality:

$$\langle F(x^\star), x - x^\star \rangle \geq 0 \qquad \text{for all } x \in X.$$

The project should expose assumptions explicitly rather than hiding them inside tactics or overly general definitions.

Core deliverables.

1. **Convex and VI infrastructure.** Define feasible sets, pseudo-gradients, dual pairings, variational inequality solutions, monotone operators, and simple gap or merit functions in a finite-dimensional setting.
2. **Bregman geometry.** Define regularizers and Bregman divergence

   $$D_\psi(u, v) = \psi(u) - \psi(v) - \langle \nabla\psi(v), u - v \rangle,$$

   then prove the algebraic identities needed for mirror descent, especially the three-point identity under explicit differentiability assumptions.
3. **Mirror-step interface.** Formalize mirror descent either as an actual argmin update when convenient, or as an abstract update satisfying the mirror optimality inequality. The abstract interface is acceptable if it produces a cleaner and more reusable theorem.
4. **Main theorem.** Prove a finite-horizon mirror descent regret bound for convex online losses, or a Lyapunov/descent inequality for a monotone variational inequality. The proof should make the Bregman telescope explicit.
5. **Examples.** Instantiate the theory for Euclidean quadratic regularization. If feasible, add a finite-dimensional simplex or entropy-style example.

6. **Stretch goal.** Formalize a mirror-prox/extragradient one-step inequality or a simple strong-monotonicity convergence result.

Expected repository structure.

```
LearningInGames/ConvexGame/Basic.lean
LearningInGames/ConvexGame/VariationalInequality.lean
LearningInGames/ConvexGame/Monotone.lean
LearningInGames/MirrorLearning/Bregman.lean
LearningInGames/MirrorLearning/MirrorStep.lean
LearningInGames/MirrorLearning/MirrorDescent.lean
LearningInGames/Examples/EuclideanMirrorDescent.lean
```

The repository should use lake, continuous integration, namespaces, comments explaining mathematical assumptions, and examples that compile from a clean checkout.

Six-month plan.

1. **Weeks 1–2: Lean and analysis sprint.** Set up the project. Learn the relevant `mathlib` infrastructure for real vector spaces, convexity, differentiability, finite-dimensional spaces, finite sums, and inequalities.
2. **Weeks 3–5: VI foundation.** Implement feasible sets, pseudo-gradients, VI solutions, monotone operators, gap functions, and basic examples. Keep the setting finite-dimensional unless generalization is easy.
3. **Weeks 6–9: Bregman layer.** Implement regularizers, Bregman divergence, algebraic identities, and the three-point identity. Make the assumptions visible in the theorem statements.
4. **Weeks 10–13: mirror step layer.** Implement the mirror step as either a concrete argmin construction or an abstract optimality interface. Prove the one-step mirror inequality.
5. **Weeks 14–18: finite-horizon theorem.** Prove the regret/descent theorem by telescoping the one-step inequality. Produce a theorem statement that is reusable for online learning and games.
6. **Weeks 19–21: examples.** Add Euclidean quadratic regularization and projected gradient descent as a mirror step. Add a simplex/entropy-style example if time permits.
7. **Weeks 22–24: ambition phase.** Add the stretch goal if feasible; otherwise polish documentation, simplify assumptions, remove fragile proofs, and write the thesis.

AI-assisted workflow. AI tools are part of the intended working method. They may generate candidate lemmas, suggest tactic scripts, locate existing `mathlib` results, and propose refactorings. The student remains responsible for theorem statements, mathematical explanations, and final code quality. No central theorem may be left as an axiom. If an abstract interface is used, it must be clearly stated and justified in the written thesis.

Success criteria. A strong thesis delivers a compiling Lean 4 library for Bregman and mirror-learning foundations; documented definitions of VI solutions, monotone operators, Bregman divergence, and mirror steps; a formal three-point or mirror-step inequality; a finite-horizon regret or Lyapunov theorem; and at least one complete example. An excellent thesis adds mirror-prox or strong-monotonicity infrastructure.

Ambition and feasibility. The thesis is ambitious because it builds a reusable optimization layer for verified learning in games. The core theorem is chosen to be foundational rather than maximally general: it should be strong enough to matter, explicit enough to formalize, and useful enough that later projects can build on it.

Prerequisites. The student should be comfortable with convex analysis, optimization, normed vector spaces, and precise theorem statements. Prior Lean 4 experience is helpful but not essential for a mathematically strong student using AI-assisted proof engineering responsibly.