Contextual bandits as a verified specification

A Z3 model of the decision layer, ported to Haskell

Matheus Hoffmann ORCID

November 2026 · Preprint v0 (draft, not peer reviewed) · rev. f5fd12d

Abstract. A contextual bandit is split into two layers: a probabilistic model that reports a mean and an uncertainty for each arm, and a decision rule that turns those estimates into a probability for each arm. We specify the decision layer in Z3 and port it to Haskell. For every real value of the estimates and for up to five arms, Z3 finds no counterexample to the following statements: greedy, UCB and epsilon-greedy return probability distributions; epsilon-greedy gives every arm at least epsilon/K, and with epsilon = 1 it is the uniform allocation of an A/B/n test; greedy and UCB pick a maximiser of their score; and the UCB optimism lemma, the deterministic core of UCB and LinUCB regret proofs, holds: with a calibrated model the arm UCB picks loses at most 2 beta w against the best arm. These are solver checks, not proofs in a proof assistant. No proof language is adopted yet; the author leans to Agda, closer to Haskell than Lean 4, used together with Z3. The probabilistic parts of a regret bound are left open.

Keywords: contextual bandits; LinUCB; Z3; formal specification; A/B testing.

1 Introduction

A contextual bandit selects one action for each request, observes a reward for that action only, and updates its model. Li, Chu, Langford and Schapire applied this model to personalized news article recommendation (Li et al. 2010).

This repository splits the agent into two layers. A probabilistic model reports a mean and an uncertainty for each arm. A decision rule turns those estimates into a probability for each arm. The Z3 specification in `z3/` proves properties of the decision rules, for every real estimate and up to five arms, and `haskell/` ports them. Z3 is used for now, and no proof assistant is adopted yet (Section 2); an earlier Lean 4 package was removed in November 2026. Section 3 lists what is checked.

2 No proof language is adopted yet

This paper does not adopt a proof language. Z3 is a solver: it checks each statement by a search for a counterexample, and it gives no proof term that a reader can check again. The checks in Section 3 are therefore Z3 checks, not proofs in a proof assistant.

The choice of a proof assistant is open. The owner is inclined to use Agda together with Z3. Agda is a dependently typed language with a syntax close to Haskell, and the owner knows Haskell better than Lean 4. Thus an Agda specification would be nearer to the Haskell port in `haskell/`. The Lean 4 package of this model was removed in November 2026. A first Agda port, `agda/`, proves the decision-layer theorems of Section 3 for each number of arms, by induction over exact rationals. Agda 2.8.0 with its standard library 2.3 checks it with `--safe`, which refuses a statement without a proof. Z3 stays the specification until the owner adopts a proof language.

3 What is proved

Table 1: Three of the statements in `z3-scala/src/main/scala/bandit/domain/Theorems.scala`.

Theorem	Statement
`epsGreedy_valid`	-greedy returns a probability distribution for $0 \leq \varepsilon \leq 1$
`epsGreedy_one_is_ab_test`	-greedy with $\varepsilon = 1$ is the uniform allocation of an A/B/n test
`ucb_gap`	with a calibrated model, UCB's arm is within $2\beta \cdot w$ of the best arm

Table 1 is a sample of the list in the repository's README. Mechanised work on stochastic bandits in Lean is in progress elsewhere (Degenne 2026).

References

Degenne, Rémy. 2026. "Bandit Algorithms in Lean (Blueprint)." Project blueprint. https://remydegenne.github.io/lean-bandits/blueprint.

Li, Lihong, Wei Chu, John Langford, and Robert E. Schapire. 2010. "A Contextual-Bandit Approach to Personalized News Article Recommendation." In *Proceedings of the 19th International Conference on World Wide Web*. https://doi.org/10.1145/1772690.1772758.