

APART × ATLAS — SECURE PROGRAM SYNTHESIS HACKATHON · TRACK 2

DiffSpec-PBT

Validating LLM-generated formal specifications via differential property-based testing. Cross-model diversity is the oracle.

• **Fatimah Emad Eldin** · Independent AI Researcher •

THE PROBLEM

Proofs are cheap. Specs are not.

LLMs can generate formal pre/post-condition specs. No-one can tell if a generated spec actually captures intent. Verify the wrong spec and you launder a bug into a theorem.

- SPOTs / SMT-based validation works but requires decidability and a second spec — re-introduces the same bottleneck.
- Feng & Sergey 2026 show PBT is unreasonably effective at validating *human-written* Lean specs.
- Our move: use PBT on *LLM-generated* specs — and use a *second LLM's* spec as the oracle.

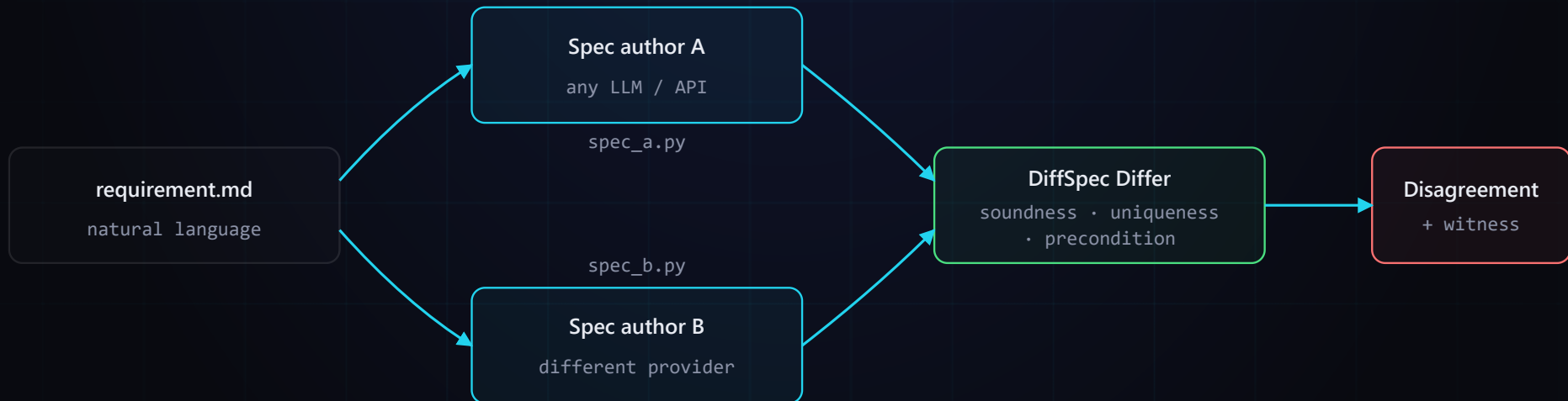
INSIGHT

Different LLMs disagree. PBT shows you where.

Prompt two independent spec authors with the same natural-language requirement. They produce different pre/post-conditions. Property-based testing surfaces a concrete input or output on which they disagree — a witness of intent drift. Works with any pair of OpenAI-compatible inference endpoints.

No SMT. No Lean. No golden reference. No manual proof effort.

One requirement, two specs, three checks.



- **Provider-agnostic:** OpenAI-compatible REST; swap models without changing the differ.
- **N-way eval:** run every pair among N authors — drift on *any* pair flags the task.
- **Fixture mode by default** — headline numbers reproduce with no API key.
- **Live mode** regenerates fixtures via `bash scripts/gen_fixtures.sh`.

THREE CHECKS

Each disagreement carries its own meaning.

① SOUNDNESS

Sample input i , compute $o = \text{impl}(i)$. If $\text{post_A}(i, o) \neq \text{post_B}(i, o)$, the rejecting spec is **OVER**-specified.

② UNIQUENESS

Sample input i and *mutated* output o' . If $\text{post_A}(i, o') \neq \text{post_B}(i, o')$, the accepting spec is **UNDER**-specified.

③ PRECONDITION

Find i where $\text{pre_A}(i) \neq \text{pre_B}(i)$. The specs disagree on what counts as a valid input.

Each disagreement is auto-classified into one of five types: **OVERSPEC**, **UNDERSPEC**, **PRECOND_MISMATCH**, **EDGE_DRIFT**, **CRASH**.

Cross-model diversity replaces the golden reference.

```
# Requirement: truncate a token sequence  
# to at most N tokens.
```

```
# Stricter spec (prefix-only):
```

```
def post(i, o):  
    tokens, n = i  
    expected_len = min(n, len(tokens))  
    return o == tokens[:expected_len]
```

```
# Weaker spec (any subsequence ≤ n):
```

```
def post(i, o):  
    tokens, n = i  
    return len(o) <= n and \  
        all(x in tokens for x in o)
```

```
# PBT finds: input=([0], 1), o'=[]
```

```
# weaker accepts []; stricter rejects
```

```
# → UNDERSPEC in 46 examples
```

- We **never need the right spec**. We just need two different wrong ones.
- Different LLMs have different training-data blind spots, so they tend to make **different** mistakes.
- Hypothesis shrinks each counterexample to a **minimal** witness — the diagnosis is auditable.
- The only thing the methodology requires is that both specs be **executable Python**.

Limitation: **correlated failure**. If every author encodes the same wrong belief, the differ stays silent — run more providers and more pairs; mutation testing of the differ is future work.

BENCHMARK

11 tasks: textbook, security, real-world.

ID	TASK	CLASS	NOTES
01–05	sort · dedup · binary_search · kth_smallest · merge_sorted	algorithmic	methodology validation
06	tokenizer_truncate	security	first-N tokens, no reorder
07	prompt_priority	security	system → user → retrieved order
08	permission_check	security	allow-list and active flag
09	rate_limit	security	window-rollover handling
10	path_sanitize	security	strip . . , abs path, null byte
11	brd_tiebreaker	real-world	Vamos Argentina BRD §III.8, verbatim

Task 11 is the methodology's load-bearing test: a real Business Requirements Document, transcribed unmodified into the benchmark.

author A ↔ author B (baseline)	6 / 11	08 / 16
author A ↔ author C	4 / 9	
author A ↔ author D	5 / 9	
author B ↔ author C	1 / 9	
author B ↔ author D	3 / 9	
author C ↔ author D	1 / 8	

Our run used four OpenAI-compatible endpoints (two via NVIDIA NIM, two via Groq). Pair labels in `evals/results.json` map to committed fixture aliases.

UNDERSPEC × 2

tokenizer_truncate · brd_tiebreaker

OVERSPEC × 1

path_sanitize

PRECOND_MISMATCH × 3

kth_smallest · rate_limit · prompt_priority

CASE STUDY — TOKENIZER TRUNCATE

UNDERSPEC found in 46 PBT examples.

```
● ● ● $ PYTHON -M EVALS.RUN_EVALS --TASK 06_TOKEN
```

```
$ python -m evals.run_evals --task 06_tokenizer_truncate
```

task	drift	type	pbt	time
06_tokenizer_truncate	YES	UNDERSPEC	46	0.32s

```
[uniqueness] strict=False vs weak=True
```

```
input  : ([0], 1)          # one token, asking for ≤1 token
```

```
output : []                # mutated: empty result
```

Weaker spec accepts the empty truncation ($\text{len}(o) \leq n$ vacuously).

Stricter spec rejects it (requires the actual prefix).

→ Weaker spec admits arbitrary token-dropping,
not just prefix-trimming – prompt-injection adjacent.

CASE STUDY — REAL-WORLD BRD

The methodology earns its keep.

Requirement (verbatim from *Vamos Argentina* BRD §III.8):

"If two or more fans have identical scores, the tie is broken by, in order:

- (1) higher total Gameplay points,*
- (2) more unique platforms shared,*
- (3) earlier submission timestamp."*

Three-tier tiebreaker. One model encodes all three rules. The other encodes only the primary `total_score` tier.

[uniqueness] UNDERSPEC

```
input : [(0, 0, 0, 1_700_000_000),
         (0, 0, 0, 1_700_000_000)]
output: [(0, 0, 0, 1_700_000_000),
         (0, 0, 0, 1_700_000_001)]
strict: False (timestamp order wrong)
weak  : True  (primary key matches, ok)
```

→ **Weaker spec silently dropped**
the **gameplay-points** and
platform-count tiebreaker rules.
Discovered in 209 PBT examples.

Where the drift hides.

ID	TASK	DISAGREEMENT TYPE(S)	PBT TO FIRST
01	sort	— (specs agree)	—
02	dedup	— (specs agree)	—
03	binary_search	EDGE_DRIFT	106
04	kth_smallest	PRECOND_MISMATCH	170
05	merge_sorted	— (specs agree)	—
06	tokenizer_truncate	UNDERSPEC	46
07	prompt_priority	PRECOND_MISMATCH	112
08	permission_check	— (specs agree)	—
09	rate_limit	PRECOND_MISMATCH	99
10	path_sanitize	EDGE_DRIFT, OVERSPEC	33
11	brd_tiebreaker	UNDERSPEC	209

True negatives matter: 3/11 tasks had no drift on any evaluated pair. A diagnostic that flagged every pair would be useless noise.

COMPARISON

Where DiffSpec-PBT sits.

APPROACH	MECHANISM	SOUNDNESS	COST	GROUND TRUTH NEEDED?
SPOTs (Lahiri 2026)	SMT-based small proofs	formal	seconds–minutes	yes (a second spec)
Feng & Sergey 2026	PBT on human-written specs	refutation-based	seconds	yes (the spec author's intent)
Rao DiffSpec 2024	differential testing of <i>code</i>	refutation-based	seconds	yes (trusted spec)
AutoSpec / SpecGen	LLM spec generation	none	seconds	none (but no validation either)
DiffSpec-PBT	PBT on cross-LLM spec pairs	refutation-based	~1 s / task	none — diversity IS the oracle

IMPLEMENTATION

1,500 lines of Python. No SMT. No Lean.

- **Language:** Python 3.10+ — dataclasses, type hints.
- **PBT:** hypothesis with shrinking.
- **LLM access:** any OpenAI-compatible endpoint; our eval used four providers (NIM + Groq).
- **Tests:** pytest + property-based tests on the differ itself.
- **Deps:** hypothesis only at runtime. Live mode adds requests.

```
# Reproduce from a fresh clone:
$ git clone <repo>
$ cd DiffSpec-PBT-Apart-Research
$ pip install -e ".[test]"

$ make test      # unit + property-based tests
$ make eval      # 7/11 drift (any pair), ~22s

# Regenerate fixtures (provider keys in .env):
$ bash scripts/gen_fixtures.sh --task 06_token --model <alias>

# Fixture aliases in repo: qwen, llama, gpt-oss, compound
```

L1 · CORRELATED FAILURE

If every author makes the same wrong choice, we stay silent. Two authors is the minimum; more providers and pairs help.

L2 · EXECUTABILITY

Both specs must run as Python. Non-decidable but executable specs are fine; specs needing a prover are out of scope.

L3 · GENERATION NOISE

Even at temperature=0.0, some providers return truncated or invalid Python. Fixture mode insulates the eval; live mode must retry.

L4 · STRATEGY AUTHORIZING

Input strategies are hand-written. A weak strategy misses real disagreements. Future: LLM-authored strategies validated by the same differ.

L5 · NO AUTOMATIC REPAIR

We surface the drift; we don't fix it. The taxonomy type points to a repair heuristic — agent-loop work for future iteration.

L6 · PAIRWISE ENGINE

1 · LEAN PORT

Re-implement on top of Lean's `Plausible` so validated specs flow directly into proof obligations.

2 · MORE PROVIDERS + MAJORITY VOTE

Add Mistral, DeepSeek, Nemotron, etc. Aggregate pair-wise witnesses with majority vote; flag all-agree-and-wrong triples.

3 · MUTATION TESTING THE DIFFER

Mutate the specs and verify the differ catches the mutation. Like mutation testing of test suites, applied to spec validators.

4 · AUTOMATIC SPEC REPAIR

Each taxonomy type points to a repair heuristic (`UNDERSPEC` → add a conjunct; `OVERSPEC` → weaken one). Agent-loop closes the gap.

5 · CI INTEGRATION

Drop-in GitHub Action that runs DiffSpec-PBT on every PR modifying a spec. Sub-second per task means it fits the per-PR budget.

6 · LLM AUTHORED STRATEGIES

THANK YOU

Cross-model diversity, made auditable.

Specifications are the bottleneck in trustworthy AI software. DiffSpec-PBT validates LLM-generated specs against each other, in seconds, without ever knowing the right answer.

7/11

TASKS FLAGGED

64%

DRIFT RATE (4-WAY)

~7 s

END-TO-END

MIT

LICENSED, CLONE TODAY