

§ 14 PoC 11 — 「reject 対象」非決定論カテゴリの実測 (reject せずに走らせて実態確認)

項目	値
対象 spec	spec_section_14_draft.md v0.1 § 14 / reject ポリシー再検討
実施日	2026-06-04
環境	WSL2 Ubuntu 24.04 / CPython 3.12.3 / RustPython 0.5.0 / PYTHONHASHSEED=0
目標	§ 14 で「本質的に非決定論」として reject されている各カテゴリを、 reject せず実際に走らせて 、(1) 同一 CPython で run-to-run に決定論か、(2) CPython vs RustPython で一致するか、(3) Isolate 制約(seed 注入/clock 固定)で決定論化するか、を実測。reject ポリシーを断定でなく 測定 で根拠づける
結果	「非決定論=一律 reject」は粗すぎると判明。実態は 4 クラス に分かれる(下記)。特に seed 注入後の random は CPython=RustPython で完全 bit-exact 、 gc は実装間で乖離、C 拡張は Full tier 専用

0. 結論

reject 対象の 7 カテゴリを実機で走らせた結果、「本質的に非決定論」は単一概念ではなく、**観測可能な 4 クラス**に分解された:

- Isolate 制約で決定論化(むしろ bit-exact):** seed なし **random.***、value-significant **time.***。
Isolate が seed 注入 / clock 固定すれば決定論。**random** は **seed 注入で CPython = RustPython** ま
で**一致**(実装跨ぎ bit-exact)。
- 実装内は決定論・実装間で乖離:** **gc.get_count()** は CPython 内で安定だが CPython ≠
RustPython。
- Full tier 専用:** **numpy** 等 C 拡張は CPython Isolate で動くが **RustPython には無い**。
- 真に非決定論 / 正規化困難:** **id()** (アドレス)、保証なしの真並列 **threading**。値の決定論化は無意
味 → reject か canonicalize か opt-in。

→ reject ポリシーは「一律拒否」でなく、この **measured spectrum** に沿って tier / determinize / opt-in
を出し分けるのが正しい。

1. 方法

各カテゴリ 1 サンプルを `engine/poc11_reject/samples/` に用意し、**reject せず素で実行**: - `python3` を 2 回(同一実装の run-to-run 決定論) - `rustpython` (実装間の一致 / 可用性) - `time` と `threading` は初回の probe が粗かったため**鋭い probe** で測り直し(全精度時刻 / `interleave switch` 数)

正しさの物差しは従来同様 `stdout` の一致。

2. 実測結果

サンプル	CPython 2回	CPython vs RustPython	観測
<code>01_random_noseed</code> <code>randint×3</code>	DIFFER	DIFFER	真に非決定論
<code>02_random_seeded</code> <code>seed(42)</code>	SAME	SAME(<code>686579303,119540831,26855092</code>)	bit-exact ・実装跨ぎ一致
<code>03_time_value</code> <code>int(time())</code>	(見かけ SAME)	—	1 秒丸めで仮装、§ 3 参照
<code>03b_time_float</code> <code>repr(time())</code>	DIFFER(<code>...94.2886</code> vs <code>...94.3046</code>)	—	壁時計依存・非決定論(確定)
<code>04_id_object</code> <code>id(),id()</code>	DIFFER	DIFFER	非決定論(アドレス)
<code>05_threading_race</code>	SAME	SAME	§ 3 参照(粗い probe)
<code>05b_thread_interleave</code> switch数	SAME(×3)	SAME	GIL+短タスクで実測決定論(保証なし)
<code>06_gc_state</code> <code>get_count()</code>	SAME	DIFFER(<code>(554,8,0)</code> vs <code>(1248,1880,0)</code>)	実装内決定論・実装間乖離
<code>07_cext_numpy</code>	SAME	ERR: No module ‘numpy’	Full tier 専用

3. 正直な測定上の注意(AI 罠 17 回避)

- **time**: 初回 `int(time.time())` は秒丸めで 2 回とも同値に見えたが、**仮装**。全精度 `repr(time.time())` で測り直すと run-to-run で割れる(`...94.2886` vs `...94.3046`)= 壁時計依存の非決定論で確定。

- **threading**: 初回 probe(len/先頭/末尾)は割れず、鋭い probe(switch 数)でも CPython×3・RustPython と一致 = 今回の測定では決定論。ただしこれは GIL + 短タスクで偶発的にスレッドがほぼ逐次実行されたため。負荷増・`sys.setswitchinterval` 変更・free-threading ビルドでは割れ得る → 「決定論は保証されない」が正しい表現。実測で決定論だった事実と、保証できない事実を両方記載する。
- **id / gc**: `id` は同一 run 内で 2 値が一致して見えるのは先行 object が GC 回収されアドレス再利用されたため。`gc.get_count()` は実装内安定だが内部構造が違う CPython/RustPython で当然乖離。

4. measured spectrum → ポリシー含意

クラス	カテゴリ	扱い(提案)
A. Isolate で決定論化	seed なし <code>random</code> 、value <code>time</code>	Isolate が seed 注入 / clock 固定 → 決定論化。値は「正規化された合成値」と明示警告。 <code>random</code> は seed 注入で実装跨ぎ bit-exact(強い)
B. 実装間乖離	<code>gc.*</code>	Full tier + CPython pin 限定。Light(RustPython)では乖離
C. Full tier 専用	C 拡張(numpy/cython/cffi/ctypes)	Full tier(CPython Isolate)のみ。Light 不可
D. 真に非決定論	<code>id()</code> 、保証なし真並列 <code>threading</code>	デフォルト reject。opt-in 時は bit-exact 保証外(best-effort)を機械可読に明示。 <code>id</code> は値を露出しない canonicalize 推奨

これにより、ユーザー選択制ポリシー(デフォルト安全 + 明示 opt-in + 警告)が、**断定でなく実測に裏付けられる**。「非決定論を一律 reject」から「測って 4 クラスに出し分け」へ。

5. 配布物

```
engine/poc11_reject/
└─ samples/ 01_random_noseed / 02_random_seeded / 03_time_value / 03b_time_float /
              04_id_object / 05_threading_race / 05b_thread_interleave / 06_gc_state /
07_cext_numpy
```

再現:

```
cd engine/poc11_reject
for f in samples/*.py; do python3 "$f"; python3 "$f"; rustpython "$f"; done # 2回
+rustpython で割れ/一致を観察
```

6. § 14.9 PoC ロードマップ 進捗

Phase	状態
☒PoC 0-10	実証 / Full tier / Light tier+WASM / 自動振り分け / 動的 HOT 判定
☒PoC 11	reject 対象非決定論カテゴリの実測 → 4 クラス spectrum
PoC 12	外部 I/O(open / requests / subprocess / os.environ)実測 / random seed 注入の実装跨ぎ 網羅 / opt-in 機構(--allow-non-deterministic / # slime: allow=)の PoC

PoC 11 完了。reject 対象の非決定論カテゴリを reject せず実走させ、「本質的に非決定論」が単一でなく 4 クラス(Isolate で決定論化 / 実装間乖離 / Full tier 専用 / 真に非決定論)に分かれることを実測。seed 注入後 random は CPython=RustPython で完全 bit-exact、gc は実装間乖離、C 拡張は Full tier 専用、id / 真並列は真に非決定論。time の秒丸め仮装と threading の GIL 依存決定論は正直に注記。reject ポリシーは「一律拒否」から「measured spectrum に沿った tier/determinize/opt-in の出し分け」へ——ユーザー選択制ポリシーを実測で裏付けた。