

FalseGreen Verification Report

ACCEPTED

All required criteria passed for the exact source tree built and verified by FalseGreen.

This does not prove universal correctness. Assurance is limited to the explicit formal properties, assumptions, models, summaries, and scope recorded in this report.

THE WORK

Job	645a6360-4a48-4323-9cc6-aaf06d4a706f
Replaces (supersede)	301e97ff-620a-43c5-981c-57cf719b8c5d — terminal verdict: accepted
Language	rust
Toolchain	rustc 1.99.1 (rustc) (1.99.1)
Verifier environment	rust - rustc 1.99.1 (rustc) (1.99.1)

THE FROZEN BOUNDARY

Source tree	70708a48e95a2a0f015e5eec37af6d1498b0c9feade57d0e39c6dd9de4529b4b
Contract SHA-256	93cb341daba1ef96c3fee2f102e03244c355b2d1dfe6f8207b186ffc3cbf4919
Criteria SHA-256	04b8ab0a890de6a56a6403723bf51a2a46c5be704f5cfb3a85eaac89bc0528b3
Criteria	4 / 4 PASSED

WHAT FALSEGREEN DID

Attempts	1
Duration	1m 33s
Report schema	v13
Report record	1 (the verdict mint)
Signed	2026-10-28 01:03:00 UTC

Contract (frozen)

Wall budget 2700s

Criterion	Title	Required	Result	Duration
build	the program compiles (release)	yes	PASS	21.4s
tests	the full test suite passes offline	yes	PASS	9.2s
clippy	clippy-clean (-D warnings)	yes	PASS	5.6s
formatting	rustfmt-clean	no	PASS	0.1s

Invariants chain linking is exact successor arithmetic and never wraps at the u64 ceiling
— formal-proofs, floor-green, both-rungs-proven

Plan (frozen)

Objective	Verify the open-source falsegreen-ledger program exactly as published: the mechanical floor (release build, full offline test suite, clippy -D warnings, rustfmt) as frozen criteria, and one frozen claim — chain linking is exact successor arithmetic and never wraps at the u64 ceiling — carried by two formal encodings the machines discharge (Kani bounded over the shipped src/rules.rs arithmetic; an unbounded Creusot Why3 proof of the guarded Pearlite spec in src/spec.rs over both translated items).
Plan digest	072290bc3d6605b2eb4a02abfc6f152c485f412abc5268815dd23477e1bfa8fd
Work items	2

Work items, in dependency order:

Work item	Outcome	Depends on	Omission risk
mechanical-floor	the program builds, tests, lints and formats clean offline, exactly as published	-	a proof-only claim carried without the mechanical floor would leave the shipped binary itself unverified
formal-proofs	the chain-link claim is discharged by both formal encodings on the published source	mechanical-floor	the encodings encode the claim as read at freeze time — the encodingclaim equivalence is the frozen human judgement the report exposes verbatim

Acceptance conditions, and the evidence that decides each:

Scope	Condition	Evidence
mechanical-floor	every frozen criterion passes on the published source: release build, the full test suite, clippy with -D warnings, rustfmt check	build, behavioral-test, static-analysis, build
formal-proofs	the Kani rung returns proven over the shipped arithmetic (no counterexample), and the Creusot rung closes every verification condition for the frozen targets (measured: 4/4 obligations, incl. the shipped rule's u64-addition safety obligation)	proof, proof
overall	the published program is accepted: all four criteria green AND both proof rows proven, in one attempt, with the per-obligation prover disclosure and verbatim clauses in the signed report	behavioral-test, proof

Evidence manifest

ATTEMPT 1 - PASSED - 2026-10-28 01:03:00 UTC - WALL 1M 33S

source sha256: 70708a48e95a2a0f015e5eec37af6d1498b0c9feade57d0e39c6dd9de4529b4b

env sha256: a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8

Security Findings

The bar these findings are judged against (the frozen policy): **fail on:** critical, high · **unknown severity:** insufficient_evidence.

Nothing to act on — 177 recorded facts (are inventory evidence in the signed JSON report).

The full SBOM and VEX are in the signed JSON report.

JSON report: <https://falsegreen.com/report/645a6360-4a48-4323-9cc6-aaf06d4a706f/json>

Invariant Checks

Each entry is ONE frozen claim, the PROVER that worked on it, and the formal proposition that prover was asked to discharge — read out of the frozen encoding, so the machine's text sits next to the author's sentence. The full rows (any counterexample trace, the admitted assumptions) are in the signed JSON report.

ATTEMPT #1

chain linking is exact successor arithmetic and never wraps at the u64 ceiling

PROVEN

"record N links to record N-1 iff prev_hash equals record N-1's hash AND seq is the exact u64 successor of record N-1's seq; the successor is never taken modulo the u64 ceiling — no seq may link past u64::MAX (the chain cannot wrap) — and on the guarded domain (prev below u64::MAX) the predicate is exactly $\text{seq} == \text{prev} + 1$ "

user-provided description — not machine-checked

Formal claim:

```
kani::assume(seq < u64::MAX)
```

```
kani::assert(!wrapped)
```

```
kani::assume(prev < u64::MAX)
```

```
kani::assert(got == (seq == prev.wrapping_add(1)))
```

Harness: **successor_never_wraps_at_the_u64_ceiling**

encoding sha256: e87bb002ddad5cf216c2e7df97470f826a80205820eff0e66ffd16bcf5c19d24

decided by: kani 0.68.0 (bounded)

chain linking is exact successor arithmetic and never wraps at the u64 ceiling

PROVEN 4 / 4

"record N links to record N-1 iff prev_hash equals record N-1's hash AND seq is the exact u64 successor of record N-1's seq; the successor is never taken modulo the u64 ceiling — no seq may link past u64::MAX (the chain cannot wrap) — and on the guarded domain (prev below u64::MAX) the predicate is exactly seq == prev + 1"

user-provided description — not machine-checked

Formal claim:

```
#[ensures(result == (seq == prev + 1u64))]
```

```
#[requires(prev < 18446744073709551615u64)]
```

Target: **links**

encoding sha256: 92ba29b838925b5024384e389e748475d9bb4e41d1df99beb791ab953efcad62

provers: alt-ergo closed Coma::vc_checked_add (5 ms); alt-ergo closed Coma::vc_elim_Some (6 ms); alt-ergo closed Coma::vc_links (10 ms); alt-ergo closed Coma::vc_links (14 ms)

decided by: creusot v0.14.0 (65349025c67e) (unbounded) on alt-ergo

Security

Policy	falsegreen-platform
Policy digest	372d3c74758acb113c1bef14b20b08b326a7703dd57d943247de6541743c0fad
Result	SATISFIED

Evidence class	Required	Status	Reasons
Dependency/SCA	yes	PASS	
Static analysis	yes	PASS	
Secrets	yes	PASS	
Supply chain	yes	PASS	89 components inventoried (evidence, in the signed JSON report)
License policy	yes	NOT ASSESSED	no licence policy frozen — the licence facts are inventory, nothing was asserted · 88 licence facts inventoried (evidence, in the signed JSON report)

How to read this table. **PASS** — the evidence was collected and nothing in it violated the frozen policy. **FAIL** — the evidence showed a violation the policy forbids; that alone fails the Job. **INSUFFICIENT EVIDENCE** — the evidence could not be collected, or could not be decided for a reason outside the code (a stale snapshot, an unavailable tool, output the parser could not read). It is never a pass, and it holds the Job at *insufficient_evidence* rather than accepting it. **NOT ASSESSED** — no policy is frozen for this class, so nothing was asserted about it; the facts are recorded as inventory and the class does not hold the Job. The findings table lists only what the frozen policy **JUDGED** — a graded severity, supply-chain drift, or a licence breaking the frozen policy. Everything else the tools recorded (components, licence facts) is inventory: not a finding, counted on the class row above, kept in the signed JSON report. A finding is **OPEN** unless it is resolved, not affected, or covered by a recorded exception.

Attempt security history (the full per-attempt sections are in the signed JSON report):

Attempt	Result	Open findings	Demonstrated violations
#1	SATISFIED	0	—

Witnesses (tool provenance)

```
osv-scanner 2.7.0 (Dependency/SCA) — invocation sha256:
e65b8610d3a7e49822362588c7c0ad8323cb68763e2f8f547b33779e60c165e5 —
a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8
```

semgrep 1.181.0 (Static analysis) – invocation sha256:
6d19a3070eeab77e2d607694f3a858d21fb646fecac277f4d4c2cbd79126a487 –
a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8

gitleaks 8.30.1 (Secrets) – invocation sha256:
76a219f7fa172309f23e0ce0b82c04725354797f69f7b131c92b5f276fff7a12 –
a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8

syft 1.54.1 (Supply chain) – invocation sha256:
eef337cabfa4ffc22335196eae78d92063e72fa4e51e2ec3d2d2f97b38301a80 –
a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8

dependency-manifest 1 (License policy) – invocation sha256:
eecf84121d1881c077d7ac91a9035cb43ce332e2fe23a8307107a23f0eea96c6 –
a4a5bf077e04f6aeb04a2ea1c0b0852414491d85daa6909080ad17cbf5f7f2e8

Tools & provers used

Tool	Version	What it did in this run
osv-scanner	2.7.0	Dependency/SCA evidence witness
semgrep	1.181.0	Static analysis evidence witness
gitleaks	8.30.1	Secrets evidence witness
syft	1.54.1	Supply chain evidence witness
dependency-manifest	1	License policy evidence witness
kani	0.68.0	invariant proofs (1 row)
creusot	v0.14.0 (65349025c67e)	invariant proofs (1 row)
alt-ergo, z3	not recorded (image-pinned)	authorized provers (frozen contract)

Exports

Full content is embedded in the signed JSON report (report schema v13).

spdx-3.0 sha256:b0d1a45bd3f26e73a0d907549a605ab07a7e846d4e2352eb94d798a0b7582a3d embedded in the JSON report (66947 bytes)

cyclonedx-1.7 sha256:8a23b906c0d49c0932df91526ca4734e1a40ba84ca8d1ff29b88faba692394ec embedded in the JSON report (32209 bytes)

cyclonedx-1.7-vex sha256:bcb2993e5908166bc0f99215bf7e14bf48ad76f3081437da4e42b9f19fd8ae7b embedded in the JSON report (632 bytes)

Signature

This report is cryptographically signed: Ed25519 over the canonical report content. The public key is embedded in the report — verification is offline and registry-free (falsegreen `verify-report`).

key id: fg-report-9fc1a2c5-134c-4ca1-a0f2-56b2d1bcde67-20261018-1

ed25519 public key: ie4eQdY1lt705aCCQz0+/192Yzc4pLQsGyVNqSKtImA=

ed25519 signature: q8nHMzGUGsXVqqecllF9wZ4fJKlyAyPxUl0dW39v1YDRJFaQ35z+wtFcLaCl4WIK0jQ23izXMGZvWC9Ay95Dw==

digest: 71857148951d29c6d07e8b6d38a8dfe8c3b4b9e672957e7ef258d717891994c3

This is a presentation of the signed canonical evidence (the JSON artifact). It is not an audit certificate. - FalseGreen v0.16.0