Proof-Gap Protocol for Source-Faithful Formalization

2026-05-11

1 Purpose

This note fixes terminology for a recurring failure mode in the formalization of the low individual degree test paper, `arXiv:2009.12982`. It complements the note-writing policy in `docs/paper-gaps/policy.tex` and the source-statement audit tracked historically in GitHub issue #1458. The current source-boundary proof holes are tracked in GitHub issue #1507 and GitHub issue #422. A proof may be blocked at an intermediate mathematical assertion. If that assertion is added as a new hypothesis to a theorem labelled as a paper theorem, the resulting Lean theorem may be correct as a conditional statement, but it is no longer the theorem in the cited source. The apparent proof has not closed the paper theorem; it has moved a proof obligation into the theorem statement.

The protocol below distinguishes such proof debt from legitimate formal boundary conditions and from genuine mathematical corrections. It is meant for contributors and reviewers who must decide whether a bridge, residual, repair, input, or obligation structure is a proved local construction, a tracked proof gap, or statement drift which must be repaired before a theorem may be advertised as a source theorem. The default rule is restrictive: such data must not be added to a paper-facing theorem unless it is part of the cited source statement or a documented faithful encoding of its ambient hypotheses.

2 Basic objects

Source theorem. A source theorem is a theorem, lemma, proposition, or corollary in the paper source under `references/ldt-paper/`. Its hypotheses and conclusion are the mathematical ground truth, subject only to faithful formal encoding and to documented paper-gap corrections.

Paper-facing declaration. A Lean declaration is paper-facing when it is named or cited as the formalization of a source theorem, for instance through a blueprint `\lean{}` link on a theorem environment carrying a paper label. Such a declaration must match the source theorem in its public statement. In particular, it may not add an unproved bridge, residual, repair, input, obligation structure, or other proof-obligation hypothesis which the paper does not assume.

Proof obligation. A proof obligation is an intermediate assertion needed by the proof of a source theorem. It may be a construction, an estimate, a transport lemma, or a comparison between two formal representations. If the paper proves or invokes this assertion, then the formalization should state it as a separate lemma or theorem and prove it from the upstream hypotheses.

Obligation discharger. An obligation discharger is a theorem or definition which constructs a proof obligation from the paper hypotheses already available at that point in the argument. This is the preferred representation of missing work. If the proof is not yet complete, the discharger may temporarily contain a tracked `sorry`, but the paper-facing theorem should call it rather than assume its output as an additional theorem hypothesis.

Conditional helper. A conditional helper is a separately named theorem whose statement explicitly assumes a proof obligation. Names such as mainFormal_ofInternalObligations, orthonormalization_ofInput, or names whose suffix advertises the assumed obligation are permitted only when the declaration is not presented as the source theorem. A conditional helper is a quarantine for downstream proof content; it does not close the source theorem.

3 Bridge, residual, repair, input, and obligation structures

The words *bridge*, *residual*, *repair*, *input*, and *obligation structure* name possible formal shapes. They become dangerous when their mathematical status is unclear or when their names make unproved data look like acceptable hypotheses of a source theorem.

Bridge. A bridge should be a transport theorem between two already defined mathematical forms: for example, a tensor-placement identity, a conversion between two equivalent consistency relations, or a statement translating a source object into a formal representation. A bridge is harmless only after it is proved from previously available assumptions. It becomes proof debt when it is assumed because the transport has not yet been proved. It becomes theorem drift when the assumed bridge is added to a paper-facing theorem.

Residual. A residual is a named remainder after part of an argument has been performed. It is often a useful way to record the exact missing target after a completed construction. A residual may appear as the conclusion of a helper theorem or as the input of a quarantined conditional helper. It is not a permissible new assumption on a paper theorem unless the paper itself assumes that residual.

Input. An input is a named bundle of data or hypotheses consumed by a construction or conditional helper. It is harmless only when each load-bearing field is either part of the source hypotheses or is produced by an earlier theorem. It becomes proof debt when the bundle is merely assumed because the construction of its fields is missing. It becomes theorem drift when such a bundle is added to the public statement of a source theorem.

Repair. A repair is a mathematical construction which replaces an approximate or partial object by a better object, usually with a quantitative estimate. In the low individual degree test formalization, projectivization and orthonormalization repairs are genuine mathematical operations. A repair should therefore appear as a construction or as a theorem discharging the corresponding obligation. If a theorem assumes the output of the repair instead of constructing it, that theorem is conditional on the repair.

Obligation structure. An obligation structure is a record whose fields are the data and estimates needed by a later construction. This is acceptable only as an internal target or as a hypothesis of a conditional helper. Such a structure becomes statement drift if it is added to a paper-facing theorem instead of being constructed inside the proof. The review question is whether every load-bearing field is proved from the source hypotheses or is explicitly tracked as a missing obligation.

4 The main failure mode

Suppose the paper proves

$$H \Longrightarrow C.$$

During formalization, the proof may need an intermediate assertion B. The paper theorem should be represented as

$$H \Longrightarrow B \Longrightarrow C \quad \text{inside the proof,}$$

where B is obtained by a lemma or theorem from H. The following public statement is a different theorem:

$$H \wedge B \Longrightarrow C.$$

It is not unsound in Lean, but it is not the paper theorem. It says that the conclusion follows if the missing bridge data is supplied. Such a theorem is a conditional helper, even if its conclusion looks exactly like the paper's conclusion.

This is the situation colloquially described as "adding a bridge hypothesis". More precisely, an unproved intermediate step has been promoted from the proof body to the theorem's assumptions. The correction is not to hide the hypothesis under a different structure name. The correction is to keep a paper-facing theorem with the source statement and move the missing step to a named internal proof obligation.

5 Faithful boundary hypotheses

Not every extra formal hypothesis is a theorem drift. Some hypotheses are faithful boundary conditions introduced by formal encoding. Examples include nonnegativity needed for a real power, positivity needed to divide by a quantity, finiteness and nonemptiness of an index type, a field-model instance, or decidability instances required by Lean. Such hypotheses are acceptable when they express part of the mathematical ambient setting or make explicit a condition used implicitly in the paper.

The distinction is load-bearing. A faithful boundary hypothesis describes the domain in which the paper's objects live. A proof-debt hypothesis supplies a mathematical step that the proof should derive. If removing the hypothesis leaves an unproved construction or estimate from the paper proof, then the hypothesis is proof debt, not a boundary condition.

6 Definition drift and subtree repair

Definition drift is especially dangerous because it propagates. If a definition is changed so that it no longer denotes the object in the paper, then every downstream theorem using that definition may continue to compile while proving statements about the wrong object. Downstream theorem statements can then be repaired piecemeal, but the development remains unstable because the wrong definition is still upstream.

When a definition mismatch is found, the repair should proceed by a subtree protocol.

1. Identify the source object and the formal object precisely.

2. Freeze a source-faithful definition or state an adapter theorem relating the existing definition to the source object.

3. Recheck the downstream theorem chain against that definition or adapter.

4. Re-formalize the affected subtree from the corrected interface, rather than adding local hypotheses to each downstream theorem.

5. If a downstream proof cannot be completed, record the missing obligation at the lowest point where the paper proof needs it.

This is slower than editing a final theorem signature, but it prevents a local definition error from becoming a global theorem-statement error.

7 Allowed shapes

Shape	Allowed use	Forbidden use
Paper-facing theorem	Matches the source statement up to faithful encoding	Carries non-paper bridge, residual, repair, input, obligation-structure, or proof-obligation hypotheses
Source-shaped theorem with a tracked `sorry`	Default repair when the source proof is blocked but the statement is known	Replaced by a strengthened theorem statement merely to keep the file compiling
Conditional helper	Temporary quarantine for substantial proof content whose remaining named obligation has not yet been discharged	Introduced merely to avoid a `sorry`, or linked as the source theorem or marked as proof-complete for that theorem
Obligation discharger theorem	Constructs the obligation from paper hypotheses	Merely restates the obligation as an assumption
Input bundle	Collects produced data for a local construction	Is assumed by a source theorem instead of being produced
Obligation structure	Bundles constructed data and estimates	Hides unproved fields that are then assumed by a source theorem
Paper-gap note	Documents a real mathematical or formal-statement divergence	Justifies silent statement drift without a repair plan

8 Protocol for a blocked proof

When a source-labelled proof is blocked, use the following order.

1. Read the cited source statement and the proof passage in `references/ldt-paper/`.
2. Write the paper theorem form, the current LEAN theorem form, and the first missing intermediate assertion.

3. Decide whether the missing assertion is a faithful boundary condition, a representation adapter, an ordinary lemma from the paper proof, or a real paper-gap discrepancy.

4. If it is an ordinary lemma or adapter, create a named theorem which discharges that obligation. Do not add it to the source theorem as a new hypothesis.

5. Keep the source-labelled theorem itself in the source form, even if its proof must contain a tracked `sorry`. This is the default repair when the missing assertion has not yet been proved.

6. Create a conditional helper only to quarantine substantial proof content that already exists and would otherwise be lost. Do not create such a helper merely to avoid a `sorry` or to keep making progress. If a conditional helper remains, give it a conditional name, keep it separate from the paper-facing theorem, and record the obligation discharger that must eliminate it.

7. If the source and formal statements genuinely differ, write or update a note under `docs/paper-gaps/` before marking the statement as aligned.

8. Update the blueprint so that paper labels point only to paper-facing declarations. Conditional helpers belong in remarks or implementation notes, not in the source theorem block.

9 Retirement criterion

An auxiliary declaration should be removed from the main blueprint route, or renamed and documented as a purely internal lemma, when all three of the following conditions hold:

1. it is not a statement in the cited paper and is not an explicitly cited outside result;

2. no current proof of a paper-level target needs it as stated;

3. it assumes stronger proof data, finite-length data, completion data, repair data, synchronization data, or package fields than the paper proof requires.

If such a declaration is still useful as a formal intermediate result, it should remain a lemma or a visibly conditional helper with a planned discharge. It should not be a headline replacement for the paper theorem, and it should not support a source-labelled blueprint `\leanok` claim.

10 Review questions

A reviewer should first compare every paper-labelled declaration with the cited source statement, checking both hypotheses and conclusion up to faithful formal encoding. Bridge, residual, repair, input, obligation-structure, and proof-obligation assumptions should be absent from paper-facing theorem signatures unless they occur in the paper. If a conditional helper remains, its name should be explicitly conditional and it should not support a source theorem `\leanok` claim. Each load-bearing obligation-structure or input field should have a theorem discharging it, a tracked issue, or an explicit `sorry` at the obligation site. When a definition has changed, the downstream subtree should be checked against the corrected definition rather than patched locally. Finally, if the formal statement genuinely differs from the source statement, a paper-gap note should explain the mathematical difference and give a plan to remove or justify it.

11 Conclusion

The project should prefer visible proof obligations over invisible statement drift. A tracked `sorry` in the source-shaped theorem, or in the lowest named internal obligation needed by its proof, is less misleading than a source theorem whose assumptions have been strengthened. Conditional helpers are not a normal substitute for unfinished proofs; they are temporary quarantines for substantial proof content, and they must remain conditional until the missing obligation is discharged. The paper-facing declaration is reserved for the theorem proved by the paper, or for a documented corrected version whose mathematical difference is explicit.