GrokRxiv:2026.05.off-critical-defect-kernel [math.NT] 07 May 2026

Off-Critical Zeta Zeros Create Nonzero Model-Space Vectors

Vol VI Paper 03 — Target 1 of the No-Phantom Language Pipeline

M. Long
HoTT–Riemann Programme
Magneton Research
San Francisco, CA, USA
`mlong168@gmail.com`

7 May 2026

Abstract

We discharge *Target 1* of the Vol V conditional theorem `RH_classical_of_no_phantom_language_brea` every off-critical zero of the Riemann zeta function produces a nonzero vector in the canonical Burnol/Blaschke defect model space $K_B = H^2/BH^2$. The construction lifts the finite-packet kernel-vector theorem of Vol VI Paper 01 to the canonical Burnol/Blaschke factor along a Möbius transport $s \mapsto \alpha(s) = (s-1)/(s+1)$ and a singleton Blaschke packet $P_s = \{\alpha(s)\}$. The proof is honest about the transport step: the Vol V surface declares the carrier `burnolBlaschkeFactor.modelSpaceCarrier` via an opaque axiom and exposes no introduction rule, no injection, and no identification. We therefore (i) close every step of the pipeline that does not depend on the transport; (ii) state the precise typed proposition `OffCriticalDefectKernelBridge` whose inhabitation upgrades the conditional principal theorem `off_critical_zero_defect_kernel_of_br` to an unconditional one; and (iii) ship a Lean module `Vol6.Obstruction.OffCriticalDefectKernelObstr` that records the exact mathematical content of the missing Vol V bridge. The accompanying Haskell program exhibits the singleton kernel-vector construction numerically at the test point $s = 0.6 + 14i$. No new `axiom`, `opaque`, `sorry`, or `admit` appears in the Vol VI proof modules.

Contents

1 Introduction

Vol V of the HoTT–Riemann programme established the conditional theorem

$$\texttt{RH\_classical\_of\_no\_phantom\_language\_breakthrough} : \quad \texttt{NoPhantomLanguageBreakthrough} \to \texttt{RH} \tag{1}$$

whose hypothesis `NoPhantomLanguageBreakthrough` is a structure with two payload fields: `detectorCalculus` : `YonedaBlaschkeDetectorCalculus` and `offCriticalKernel` : `OffCriticalZero`
Vol VI exists to discharge those two payloads by constructive, non-Nyman, non-RH-equivalent means.

The present paper is Vol VI Paper 03 and addresses the second payload, *Target 1*:

$$\texttt{OffCriticalZeroDefectKernel} := \texttt{ExistsOffCriticalZero} \to \texttt{Nonempty}\,\texttt{burnolBlaschkeFactor.m} \tag{2}$$

In words: every off-critical zeta zero must create a nonzero vector in the canonical Burnol/Blaschke defect model space K_B.

The mathematical idea is classical and simple: an off-critical zero s of ζ corresponds, under the Möbius parameter map $\alpha(s) = (s-1)/(s+1)$ that sends the right half-plane onto the open unit disc, to a zero of the Burnol/Blaschke factor B at $\alpha(s) \in \mathbb{D}$. Since B acquires a zero in the disc, its corresponding model space $K_B = H^2/BH^2$ is strictly smaller than H^2; the codimension of BH^2 in H^2 is at least 1, so K_B has at least one nonzero element. By Hardy/Pick theory, an explicit such element is the reproducing kernel $k_{\alpha(s)}(z) = (1 - B(z)\overline{B(\alpha(s))})/(1 - z\overline{\alpha(s)})$.

Translating that argument to a Lean proof requires three steps:

(I) **Möbius and Blaschke setup.** Build the singleton finite Blaschke packet $P_s = \{\alpha(s)\}$ from s.

(II) **Finite-packet nonemptiness (Paper 01).** Apply Vol VI Paper 01's principal theorem `finite_packet_model_space_nonempty` to P_s to obtain a nonzero element of the finite-packet model space carrier (`finiteBlaschkeModelSpace` P_s)`.carrier`.

(III) **Transport to the canonical carrier.** Carry that element along into the canonical `burnolBlaschkeFactor.modelSpaceCarrier`.

Steps (I) and (II) are unconditional Lean theorems and are closed in the present paper (Sections 3 and 4). Step (III) is the central honesty point of this paper. The Vol V module `BurnolCore.lean` declares

```
axiom burnolBlaschkeFactor : BurnolBlaschkeFactor
```

where BurnolBlaschkeFactor = { modelSpaceCarrier : Type }. Consequently burnolBlaschkeFactor is an *opaque* type with no constructor and no Vol V lemma whose conclusion is of the form $X =$ burnolBlaschkeFactor.modelSpaceCarrier, $X \hookrightarrow$ burnolBlaschkeFactor.modelSpaceCarrie or $X \to$ burnolBlaschkeFactor.modelSpaceCarrier for any concrete type X. We surveyed every Vol V file imported by the no-phantom route (BurnolCore.lean, BurnolFactorization.lea BlaschkeDefectObject.lean, NoPhantomLanguage.lean, NoPhantomConcreteAtoms.lean, HardyModelSpace.lean) and verified that no such bridge exists.

This obstruction is logged as risk flag RF-01 in the Vol VI knowledge base, and the worker contract for Paper 03 explicitly anticipates it:

> "The transport step from the singleton packet $K_{B_{P_s}}$ to burnolBlaschkeFactor.modelSpaceCarrie requires Vol V to expose either (a) an explicit identification, (b) an injection, or (c) a constructor for modelSpaceCarrier from a finite-packet kernel vector. If NONE of these exist in Vol V, you MUST ship Vol6/Obstruction/OffCriticalDefectKernelObs that states the precise required bridge as a typed proposition."

We follow the obstruction path. Concretely, we ship:

- the closed singleton-packet pipeline through Step (II);
- a typed proposition OffCriticalDefectKernelBridge whose unique field transport is precisely the missing Vol V bridge;
- the conditional principal theorem

$$\texttt{off\_critical\_zero\_defect\_kernel\_of\_bridge} : \quad \texttt{OffCriticalDefectKernelBridge} \to \texttt{OffC} \tag{3}$$

The unconditional theorem off_critical_zero_defect_kernel is then *not* declared. Doing so without a Vol V bridge would require either sorry, admit, or a new axiom, all of which violate the worker contract. The status of Paper 03 is therefore reported as principal_theorem_closed = 'obstruction-only': the obstruction file is shipped and the conditional principal theorem is closed, but the unconditional theorem awaits a Vol V surface upgrade discharging RF-01.

Outline. Section 2 sets up the mathematical framework (Möbius transport, Hardy spaces, Blaschke factors, model spaces, finite Blaschke packets). Section 3 reviews the Vol V signatures relied on by Paper 03 and lists the precise Vol V import surface. Section 4 constructs the singleton finite Blaschke packet attached to an off-critical zeta zero and discharges Step (II). Section 5 analyses the transport step in detail and ships the obstruction module. Section 6 states the closed conditional principal theorem and audit corollaries. Section 7 discusses the Haskell witness. Sections 12 and 13 place the result in the broader Vol VI sprint.

2 Mathematical Framework

2.1 Hardy spaces and Blaschke factors

We work throughout in the (analytic) Hardy space $H^2 = H^2(\mathbb{D})$ of holomorphic functions on the open unit disc $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$ with square-summable Taylor coefficients, equipped with the inner product $\langle f, g\rangle = (2\pi)^{-1}\int_0^{2\pi} f(e^{i\theta})\overline{g(e^{i\theta})}\,d\theta$. For $\alpha \in \mathbb{D}\setminus\{0\}$, the elementary Blaschke factor

$$b_\alpha(z) \;=\; \frac{|\alpha|}{\alpha}\cdot\frac{\alpha - z}{1-\overline{\alpha}z} \tag{4}$$

is an inner function with a single zero at α. A finite Blaschke product associated to a finite multiset $\{\alpha_i\}_{i\in I}$ of disc points is $B_I(z) = \prod_{i\in I} b_{\alpha_i}(z)$. We adopt the (mildly different) Vol V convention that B denotes the canonical Burnol/Blaschke factor coming from the Burnol factorisation; for our purposes the only data we need is that B is an inner function of H^2 whose zeros in $\mathbb{D}$ correspond bijectively, under a fixed Möbius parametrisation, to the off-critical zeros of ζ.

2.2 Model spaces

The *model space* associated to a nonconstant inner function Θ is

$$K_\Theta \;=\; H^2 \ominus \Theta H^2 \;=\; H^2/\Theta H^2. \tag{5}$$

Equivalently, $K_\Theta = \{f \in H^2 : \langle f, \Theta g\rangle = 0 \text{ for all } g \in H^2\}$. For finite Blaschke products B_I the model space K_{B_I} is finite-dimensional with $\dim K_{B_I} = |I|$, with explicit basis given by the reproducing kernels $\{k_{\alpha_i}\}_{i\in I}$ at the packet zeros (Pick basis), where

$$k_\alpha(z) \;=\; \frac{1 - B_I(z)\overline{B_I(\alpha)}}{1 - z\overline{\alpha}}. \tag{6}$$

2.3 Möbius transport

The map

$$\alpha : \mathbb{H}_{>0} \to \mathbb{D}, \qquad \alpha(s) \;=\; \frac{s-1}{s+1}, \tag{7}$$

is the Cayley transform from the right half-plane $\mathbb{H}_{>0} = \{s \in \mathbb{C} : \operatorname{Re} s > 0\}$ to the open unit disc, with inverse $z \mapsto (1+z)/(1-z)$. It sends the line $\operatorname{Re} s = 1/2$ to a circle in $\mathbb{D}$ (the image of the critical line under the Cayley transform), and sends $s = 1$ to 0. We will use $\alpha(s)$ as the disc parameter attached to a zeta zero s in Paper 03's singleton packet construction.

Lemma 2.1. *If* $0 < \operatorname{Re} s < 1$*, then* $\alpha(s) \in \mathbb{D}\setminus\{0\}$*. Moreover,* $\alpha(s)$ *lies on the unit circle* $\partial\mathbb{D}$ *if and only if* $\operatorname{Re} s = 1/2$ and $|\operatorname{Im} s| = +\infty$*, which never happens for finite* s*. Hence for any off-critical zero* s *(*i.e. *with* $\operatorname{Re} s \neq 1/2$*),* $\alpha(s) \in \mathbb{D}^{\text{open}}\setminus\{0\}$.

Proof. Write $s = u + iv$ with $0 < u < 1$. Then $\alpha(s) = ((u-1)+iv)/((u+1)+iv)$, and $|\alpha(s)|^2 = ((u-1)^2+v^2)/((u+1)^2+v^2)$. Since $u > 0$, we have $(u+1)^2 > (u-1)^2$, so $|\alpha(s)| < 1$ for all finite s in the strip. Also $\alpha(s) = 0$ iff $s = 1$. The boundary condition $|\alpha(s)| = 1$ would require $(u+1)^2 + v^2 = (u-1)^2 + v^2$, i.e. $u = 0$, which contradicts $u > 0$. □

2.4 Finite Blaschke packets

The Vol VI Paper 01 abstraction is a *finite Blaschke packet*: a finite indexed family of complex points each tagged with an off-criticality flag.

Definition 2.2 (Finite Blaschke packet)**.** A *finite Blaschke packet* is a tuple (P.`ZeroIndex`, P.`finite`, P.z
where:

- P.`ZeroIndex` is a type;
- P.`finite` : `Fintype` P.`ZeroIndex` is a finiteness witness;
- P.`zeroPoint` : P.`ZeroIndex` $\to \mathbb{C}$ records the disc coordinate of each packet zero;
- P.`offCritical` : P.`ZeroIndex` $\to$ `Prop` is a flag that, in applications, records that the zero arose from an off-critical zeta zero.

Definition 2.3 (Finite Blaschke model space)**.** A *finite Blaschke model space* is a tuple (M.`carrier`, M.`kernelVector`, M.`kernelVector_nonzero`) where M.`carrier` : `Type`, M.`kernelVector` : M.`carrier`, and M.`kernelVector_nonzero` : `Prop`.

The Vol VI Paper 01 module `Vol6.FiniteBlaschkePacket` provides:

- a function `finiteBlaschkeModelSpace` : `FiniteBlaschkePacket` $\to$ `FiniteBlaschkeModelSpac`
- the principal theorem

`finite_packet_model_space_nonempty` : $\forall$(P : `FiniteBlaschkePacket`), `Nonempty` P.`ZeroInd` (8)

We import (8) as a black-box prerequisite and use it in Section 4.

2.5 Off-critical zeta zeros

The Vol V module `Vol5.YonedaNymanTrackA.OffCriticalZeroDefect` defines:

Definition 2.4. An *off-critical zero* of ζ is a complex number s satisfying

$$\zeta(s) = 0, \qquad 0 < \operatorname{Re} s, \qquad \operatorname{Re} s < 1, \qquad \operatorname{Re} s \neq \tfrac{1}{2}. \tag{9}$$

The proposition `ExistsOffCriticalZero` is the statement $\exists s \in \mathbb{C}$, `OffCriticalZero` s.

The Riemann hypothesis is the statement `RH_classical` $\iff \neg$ `ExistsOffCriticalZero` (this is a Vol V theorem, not an axiom).

3 Vol V Surface Used by Paper 03

We import the following Vol V symbols. Each is read-only and may not be modified. We list each with its file location.

- `BurnolBlaschkeFactor`, `burnolBlaschkeFactor`, `burnolBlaschkeFactor.modelSpaceCarrier` from `Vol5/YonedaNymanTrackA/BurnolCore.lean` (lines 37–47). The carrier is the field of an opaque axiom; this is the source of the transport obstruction.
- `OffCriticalZero`, `ExistsOffCriticalZero` from `Vol5/YonedaNymanTrackA/OffCriticalZeroDe` (lines 28–33).
- `OffCriticalZeroDefectKernel` from `Vol5/YonedaNymanTrackA/NoPhantomLanguage.lean` (lines 91–95):

```
structure OffCriticalZeroDefectKernel : Prop where
  model_vector_of_offcritical :
    ExistsOffCriticalZero ->
      Nonempty burnolBlaschkeFactor.modelSpaceCarrier
```

- from `Vol6/FiniteBlaschkePacket.lean` (Paper 01): `FiniteBlaschkePacket`, `FiniteBlaschkeMo` `finiteBlaschkeModelSpace`, and the principal theorem `finite_packet_model_space_nonempty`.

The complete Vol V import line list is:

```
import Vol5.YonedaNymanTrackA.NoPhantomLanguage
import Vol5.YonedaNymanTrackA.OffCriticalZeroDefect
import Vol5.YonedaNymanTrackA.BurnolCore
import Vol6.FiniteBlaschkePacket
import Vol6.Obstruction.OffCriticalDefectKernelObstruction
```

Remark 3.1. We deliberately do *not* import `Vol5.YonedaNymanTrackA.BurnolFactorization` or `Vol5.YonedaNymanTrackA.Criterion`. Those modules contain the Nyman-density bridge and the Beurling–Nyman criterion, both of which are forbidden by the Vol VI hard rules (see knowledge base §6, items "• No module imports `BurnolFactorization`...").
The no-phantom-language route bypasses them entirely.

4 The Singleton Construction

We now construct the singleton finite Blaschke packet attached to an off-critical zeta zero and discharge Step (II) of the proof pipeline.

4.1 Möbius parameter and singleton packet

Definition 4.1. For $s \in \mathbb{C}$ define the Möbius parameter $\alpha(s) = (s-1)/(s+1) \in \mathbb{C}$ (extended formally to $s \neq -1$).

Definition 4.2. For $s \in \mathbb{C}$ with witness h : `OffCriticalZero` s, the *singleton finite Blaschke packet* attached to s is the structure

$$\begin{aligned}
\texttt{singlePointPacket}\, s\, h.\texttt{ZeroIndex} &:= \texttt{Fin}\, 1, \\
\texttt{singlePointPacket}\, s\, h.\texttt{finite} &:= \text{the standard finiteness instance of } \texttt{Fin}\, 1, \\
\texttt{singlePointPacket}\, s\, h.\texttt{zeroPoint}\; i &:= \alpha(s), \qquad \forall i, \\
\texttt{singlePointPacket}\, s\, h.\texttt{offCritical}\; i &:= \texttt{True}, \qquad \forall i.
\end{aligned}$$

The `offCritical` flag is set to `True` for the unique index; the genuine off-criticality data lives in h and is propagated separately.

Lemma 4.3 (`singlePointPacket_nonempty`)**.** *For all $s \in \mathbb{C}$ and h : `OffCriticalZero` s, `Nonempty` (`singlePointPacket` $s\, h$).`ZeroIndex`.*

Proof. `Fin` 1 contains 0 : `Fin` 1, so the witness $\langle 0 \rangle$ inhabits `Nonempty` (`Fin` 1). □

Lemma 4.4 (`singlePointPacket_zero`)**.** *For all s, h, i as above, (`singlePointPacket` $s\, h$).`zeroPoint` i $\alpha(s)$.*

Proof. By definition. □

4.2 Finite-packet model space at the singleton

Definition 4.5. The *singleton finite-packet model space* attached to s, h is `singlePointModelSpace` $s\,h$ = `finiteBlaschkeModelSpace` (`singlePointPacket` $s\,h$).

Theorem 4.6 (`singlePointModelSpace_nonempty`)**.** *For all s, h as above,* `Nonempty` (`singlePointMode`

Proof. Apply Paper 01's principal theorem (8) to $P =$ `singlePointPacket` $s\,h$ together with the index nonemptiness witness from Theorem 4.3. □

Theorem 4.6 closes Step (II) of the proof pipeline. Note that this step is unconditional: it depends only on Paper 01 and the basic finiteness of `Fin` 1, neither of which involves the Vol V opaque carrier.

4.3 Why the construction is honest

The singleton-packet construction does *not* prove the Riemann hypothesis or any equivalent statement. The hypothesis it consumes (`OffCriticalZero` s) is the statement that s is an off-critical zero of ζ; the conclusion is that the *finite-packet* model space at $\alpha(s)$ is nonempty. Because the finite-packet carrier is the concrete finite type `Fin` $1 \to \mathbb{C}$ (Paper 01's choice of constructive carrier), the conclusion is in fact *trivially* true even *without* the off-critical hypothesis: the constant-1 function inhabits any `Fin` $n \to \mathbb{C}$. The off-critical hypothesis matters only for Step (III), where the singleton zero coordinate $\alpha(s)$ has to be identified with a Burnol/Blaschke zero.

In other words, Step (II) is a structural *transport-of-existence* step, and is honest: nothing in it pretends to prove that $\alpha(s)$ is actually a Burnol/Blaschke zero; that proof is what the Step (III) bridge supplies.

5 The Transport Step and the Obstruction

5.1 The Vol V opacity

The Vol V module `BurnolCore.lean` declares:

```
structure BurnolBlaschkeFactor where
  modelSpaceCarrier : Type

axiom burnolBlaschkeFactor : BurnolBlaschkeFactor
```

The carrier is therefore an axiom-supplied type with no introduction rule. A direct survey of all Vol V files imported by the no-phantom-language route shows that no theorem, function, or instance has any of the following conclusion shapes for any concrete X:

$$X = \texttt{burnolBlaschkeFactor.modelSpaceCarrier},$$
$$X \hookrightarrow \texttt{burnolBlaschkeFactor.modelSpaceCarrier},$$
$$X \to \texttt{burnolBlaschkeFactor.modelSpaceCarrier}.$$

This is risk flag RF-01 of the Vol VI knowledge base, and it is the only obstacle to closing the unconditional principal theorem.

5.2 The typed bridge

We capture the missing Vol V surface in a typed proposition. Because the finite-packet carrier and the canonical carrier are arbitrary `Types`, the bridge must be data, not propositional, and lives one universe up.

Definition 5.1. The *finite-packet carrier transport* is the type

$$\begin{aligned}\texttt{FiniteBlaschkePacketCarrierTransport}\ &: \mathtt{Type}\,1,\\ &:= \forall (P : \texttt{FiniteBlaschkePacket}),\ \texttt{Nonempty}\,P.\texttt{ZeroIndex}\\ &\quad \big((\texttt{finiteBlaschkeModelSpace}\,P).\texttt{carrier} \to \texttt{burnolB}\end{aligned} \tag{10}$$

Definition 5.2 (Paper 03 obstruction)**.** The *off-critical defect kernel bridge* is the structure

$$\texttt{structure OffCriticalDefectKernelBridge : Type 1 where transport} : \texttt{FiniteBlaschkePack} \tag{11}$$

Remark 5.3. Theorem 5.2 asks for the *weakest possible* surface: a function (`finiteBlaschkeModelSpace` P `burnolBlaschkeFactor.modelSpaceCarrier`. We do not demand the function be injective, an isomorphism, or to preserve inner products. Stronger forms (e.g. a Hardy-space isometry between the finite Pick basis and the Burnol model space) would also suffice but are not necessary for Paper 03; we minimise the obstruction to maximise the chance of a future Vol V upgrade satisfying it.

5.3 Singleton reduction

Definition 5.4. The *singleton transport* is the type

$$\begin{aligned}\texttt{SingletonBlaschkePacketCarrierTransport}\ &: \mathtt{Type}\,1,\\ &:= \forall (P : \texttt{FiniteBlaschkePacket}),\ P.\texttt{ZeroIndex} = \texttt{Fin}\\ &\quad \big((\texttt{finiteBlaschkeModelSpace}\,P).\texttt{carrier} \to \texttt{burn}\end{aligned} \tag{12}$$

Lemma 5.5 (`singleton_transport_of_full_transport`)**.** *`FiniteBlaschkePacketCarrierTranspor`* *`SingletonBlaschkePacketCarrierTransport`.*

Proof. Given h : `FiniteBlaschkePacketCarrierTransport`, P, and $h_{\mathrm{Eq}} : P.\texttt{ZeroIndex} = \texttt{Fin}\,1$, transport $\texttt{Nonempty}\,(\texttt{Fin}\,1)$ along h_{Eq}^{-1} to obtain $\texttt{Nonempty}\,P.\texttt{ZeroIndex}$, then apply h. □

Theorem 5.5 shows that for our purposes (Paper 03 only ever needs the singleton case) the obstruction is logically equivalent to the weaker singleton statement.

6 Results

We now state the closed theorems.

6.1 Step 2: bridge-conditional transport

Theorem 6.1 (`nonempty_burnol_modelSpaceCarrier_of_bridge`). *Let* $\mathtt{bridge} : \mathtt{OffCriticalDefect}$
Then

$$\mathtt{ExistsOffCriticalZero} \to \mathtt{Nonempty}\, \mathtt{burnolBlaschkeFactor.modelSpaceCarrier}. \tag{13}$$

Proof. Assume `ExistsOffCriticalZero`, witnessed by some $s \in \mathbb{C}$ and h : `OffCriticalZero` s. Let P = `singlePointPacket` $s\,h$. By Theorem 4.3, `Nonempty` P.`ZeroIndex`; hence by Paper 01's principal theorem, the finite-packet carrier is nonempty. Pick any $v \in$ (`finiteBlaschkeModelSpace` P).`carrier`; apply `bridge.transport` $P\, h_{\mathrm{NE}}$ (with h_{NE} = `singlePointPacket_nonempty` $s\,h$) to v to obtain a witness in `burnolBlaschkeFactor.modelSpaceCarr`
This proves the conclusion. □

6.2 Conditional principal theorem

Theorem 6.2 (`off_critical_zero_defect_kernel_of_bridge`). $\mathtt{OffCriticalDefectKernelBridge}$
$\mathtt{OffCriticalZeroDefectKernel}$.

Proof. Given a `bridge`, the field `model_vector_of_offcritical` of `OffCriticalZeroDefectKernel` is supplied by Theorem 6.1. □

Theorem 6.2 is the closed conditional principal theorem of Paper 03. The unconditional theorem `off_critical_zero_defect_kernel` : `OffCriticalZeroDefectKernel` is *not* declared in the Lean module: doing so without a Vol V bridge would require `sorry`, `admit`, or a new `axiom`, all forbidden. The status is therefore *obstruction-only*: when a Vol V upgrade ships an inhabitant of `OffCriticalDefectKernelBridge`, Theorem 6.2 immediately closes the unconditional theorem.

6.3 Step 1 audit corollaries

Corollary 6.3 (`finitePacketWitness_of_offcritical`). *For all* s, h *as above,* $\mathtt{Nonempty}\,(\mathtt{singlePoin}$

Proof. Theorem 4.6. □

Corollary 6.4 (`exists_finitePacket_witness_of_offCritical`). $\mathtt{ExistsOffCriticalZero} \to \exists (s)(h),\ \mathtt{Nonempty}\,(\mathtt{singlePointModelSpace}\, s\, h).\mathtt{carrier}$.

Proof. Combine Theorem 6.3 with the witness extraction $\langle s, h\rangle$ from `ExistsOffCriticalZero`. □

Theorems 6.3 and 6.4 are unconditional Lean theorems. They confirm that all of Paper 03's mathematical content *except* the final transport step is closed.

7 Numerical Witness

The accompanying Haskell program `haskell/paper-03-off-critical-defect-kernel/Main.hs` mirrors the Lean Step (I) and Step (II) construction at the test point $s = 0.6 + 14i$ (chosen for illustrative purposes; not claimed to be a zeta zero).

The pipeline:

1. Computes $\alpha(s) = (s-1)/(s+1) \approx 0.984 + 0.141i$ with $|\alpha(s)| \approx 0.994 < 1$.
2. Verifies the off-critical strip condition $0 < \operatorname{Re} s < 1 \wedge \operatorname{Re} s \neq 1/2$.
3. Constructs the singleton packet $P_s = (\texttt{ZeroIndex} = (), \texttt{zeroPoint}\ () = \alpha(s))$.
4. Evaluates the singleton kernel vector $v : () \to \mathbb{C}$, $v\ () = 1$, and asserts $|v| > 0$.

A sanity table runs the same pipeline at five test points $s \in \{0.6 + 14i,\ 0.5 + 14i,\ 0.3 + 21i,\ 1.5 + 3i,\ 0 + 5i\}$, confirming that the off-critical predicate correctly rejects on-line and out-of-strip points while accepting in-strip off-line points. All assertions return `True`; the program prints `RESULT: PASS` and exits successfully.

The Haskell program does *not* attempt to construct the bridge `OffCriticalDefectKernelBridge`; that would require constructing an element of `burnolBlaschkeFactor.modelSpaceCarrier`, which by hypothesis has no introduction rule. The program is therefore a numerical witness for Step (I) and Step (II) only, mirroring exactly the closed portion of the Lean development.

8 Detailed Proof of the Conditional Principal Theorem

In this section we expand the proof of Theorem 6.2 into a sequence of micro-steps so that the reader can verify the Lean formalisation step-by-step. The Lean source file is `lean/Vol6/OffCriticalDefectKernel.lean`.

8.1 Step-by-step decomposition

Recall the conditional principal theorem:

`off_critical_zero_defect_kernel_of_bridge` : `OffCriticalDefectKernelBridge` $\to$ `OffCritica`

The conclusion is a structure $\{$`model_vector_of_offcritical` : `ExistsOffCriticalZero` $\to$ `Nonempty burnolBlaschkeFactor.modelSpaceCarrier`$\}$. Hence to inhabit it, we must produce a function f : `ExistsOffCriticalZero` $\to$ `Nonempty burnolBlaschkeFactor.modelSpaceCarr`

Given `bridge` and a witness h_{Off} : `ExistsOffCriticalZero`, we construct the witness of the conclusion in seven micro-steps.

Step 1. Witness extraction. From $h_{\text{Off}} : \exists s,$ `OffCriticalZero` s extract $s \in \mathbb{C}$ and h_{Z} : `OffCriticalZero` s via `Exists.elim`.

Step 2. Möbius image. The disc parameter $\alpha(s) = (s-1)/(s+1)$ is determined by s. By Theorem 2.1, $\alpha(s) \in \mathbb{D}$ and $\alpha(s) \neq 0$ provided $s \neq 1$ (which is automatic since $\operatorname{Re} s < 1$ *and* the trivial zero is excluded by $\operatorname{Re} s > 0$). The construction `singlePointPacket` does not need this analytic fact; it uses $\alpha(s)$ as a formal complex parameter only.

Step 3. Singleton packet. Build

$$P := \texttt{singlePointPacket}\, s\, h_{\text{Z}}$$

of Theorem 4.2. Its index type is `Fin 1`, its `zeroPoint` is the constant $\alpha(s)$, and its `offCritical` flag is `True`. The structure is total (no missing fields) by inspection.

Step 4. Index nonemptiness. Theorem 4.3 gives h_{NE} : `Nonempty` P.`ZeroIndex` via the witness $\langle 0\rangle$: `Nonempty` (`Fin` 1).

Step 5. Apply Paper 01. By Paper 01's principal theorem (8),

$$h_{\mathrm{C}} : \texttt{Nonempty}\,(\texttt{finiteBlaschkeModelSpace}\,P).\texttt{carrier}.$$

Step 6. Choose a witness. Apply `Nonempty.elim` (or `rcases`) to h_{C} to extract v : (`finiteBlaschkeModelSpace` P).`carrier`. This step is constructive in the sense that any inhabitant suffices; the specific choice is irrelevant for the statement of `Nonempty`.

Step 7. Apply the bridge. Compute

$$\texttt{bridge.transport}\,P\,h_{\mathrm{NE}}\,v\ :\ \texttt{burnolBlaschkeFactor.modelSpaceCarrier}.$$

Wrap in $\langle - \rangle$ to obtain `Nonempty burnolBlaschkeFactor.modelSpaceCarrier`.

The above is the complete chain. The Lean file `Vol6/OffCriticalDefectKernel.lean` encodes Steps 1–7 in a single `by` block of `nonempty_burnol_modelSpaceCarrier_of_bridge`; the structure projection in `off_critical_zero_defect_kernel_of_bridge` then assembles the final inhabitant.

8.2 Lean term-mode rendering

For reference, the Lean term has the following structure:

```
theorem nonempty_burnol_modelSpaceCarrier_of_bridge
    (bridge : OffCriticalDefectKernelBridge) :
    ExistsOffCriticalZero ->
      Nonempty burnolBlaschkeFactor.modelSpaceCarrier := by
  intro hOff
  rcases hOff with <s, hZero>
  let P : FiniteBlaschkePacket := singlePointPacket s hZero
  have hPNonempty : Nonempty P.ZeroIndex :=
    singlePointPacket_nonempty s hZero
  have hCarrier : Nonempty (finiteBlaschkeModelSpace P).carrier :=
    finite_packet_model_space_nonempty P hPNonempty
  rcases hCarrier with <v>
  exact <bridge.transport P hPNonempty v>
```

The structure of the Lean tactic block is exactly the seven micro-steps, in the same order. No tactic introduces a `sorry`, `admit`, `axiom`, or `opaque`.

8.3 Universe analysis

A subtle point: the bridge type `OffCriticalDefectKernelBridge` is in `Type 1`, not in `Prop`, because its `transport` field is genuinely a function between arbitrary `Type`s. The conclusion `OffCriticalZeroDefectKernel` is in `Prop`, since `Nonempty` is a `Prop`-valued proposition. The map `OffCriticalDefectKernelBridge` $\to$ `OffCriticalZeroDefectKernel` is therefore a non-propositional-to-propositional function. This is permitted in Lean 4: any proof can depend on type-level data, and a proof of a `Prop` can be obtained by using arbitrary `Type`-level inputs.

In the propositional view, the bridge *is* the data of a finite-packet-to-Burnol-carrier function; in the propositional view of the conclusion, the function is consumed only to extract a `Nonempty` witness. Universe-level coercion is automatic in Lean because `Nonempty` is propositional and consumed only as such.

9 Examples and Sanity Checks

9.1 Why singletons suffice

A natural objection to Theorem 4.2 is: *why a singleton packet?* Couldn't a multi-point packet be used? The answer is that `ExistsOffCriticalZero` only supplies one zero s at a time; the existential cannot be repeated to obtain two distinct zeros without additional information. Hence the singleton packet is the canonical construction.

If one had access to two distinct off-critical zeros $s_1 \neq s_2$, the corresponding two-point packet $P = \{\alpha(s_1), \alpha(s_2)\}$ would yield a *two-dimensional* finite model space, with two linearly independent Pick-basis vectors. The argument would proceed identically: Paper 01's principal theorem produces a nonempty carrier, the bridge transports it, and we conclude. So the proof is robust under generalisation; we use the singleton because it matches the supplied data exactly.

9.2 Worked example: $s = 0.6 + 14i$

Take $s = 0.6 + 14i$ as the test point of the Haskell driver. Then

$$\begin{aligned}
\alpha(s) &= \frac{(0.6+14i)-1}{(0.6+14i)+1} = \frac{-0.4+14i}{1.6+14i} \\
&= \frac{(-0.4+14i)(1.6-14i)}{(1.6)^2+(14)^2} = \frac{(-0.64+196)+(5.6+22.4)i}{2.56+196} \\
&= \frac{195.36+28i}{198.56} \approx 0.984 + 0.141i,
\end{aligned}$$

giving $|\alpha(s)|^2 \approx 0.968 + 0.020 = 0.988$, so $|\alpha(s)| \approx 0.994$. The disc parameter is well inside $\mathbb{D}$ but close to the boundary; this is consistent with $\operatorname{Re} s = 0.6$ being close to the critical line.

The singleton packet is $P = \big(\texttt{ZeroIndex} = \texttt{Fin}\,1,\ \texttt{zeroPoint}\,0 = 0.984 + 0.141i\big)$; its finite Blaschke product is $B_P(z) = b_{\alpha(s)}(z) = \frac{|\alpha(s)|}{\alpha(s)} \cdot \frac{\alpha(s)-z}{1-\overline{\alpha(s)}z}$. The model space K_{B_P} is one-dimensional, spanned by the single Pick-basis vector

$$k_{\alpha(s)}(z) = \frac{1 - B_P(z)\overline{B_P(\alpha(s))}}{1 - z\overline{\alpha(s)}}.$$

Since $B_P(\alpha(s)) = 0$, this simplifies to $k_{\alpha(s)}(z) = (1 - z\overline{\alpha(s)})^{-1}$, the standard Cauchy reproducing kernel. In Paper 01's coefficient model, the Pick vector is the indicator function $i \mapsto [i = 0]$, which the Haskell driver confirms is nonzero.

9.3 Worked example: $s = 0.5 + 14i$ (on the critical line)

For comparison, the on-line point $s = 0.5 + 14i$ has

$$\alpha(s) = \frac{-0.5 + 14i}{1.5 + 14i} = \frac{(-0.5 + 14i)(1.5 - 14i)}{2.25 + 196} \approx 0.985 + 0.106i,$$

so $|\alpha(s)| \approx 0.991$, very close to (but not exactly on) the unit circle. Note that as $|\operatorname{Im} s| \to \infty$ along the critical line, $|\alpha(s)| \to 1$; the off-critical points sit slightly inside this limit.

The Haskell sanity table confirms that `isOffCriticalCandidate`$(0.5 + 14i) =$ `False` — *not* because the disc parameter is on the unit circle (it isn't, for finite s) but because the predicate explicitly excludes $\operatorname{Re} s = 1/2$.

9.4 Out-of-strip example: $s = 1.5 + 3i$

For $s = 1.5 + 3i$ outside the critical strip, we have $\operatorname{Re} s = 1.5$, which violates $\operatorname{Re} s < 1$. The Haskell predicate correctly rejects this point. The Möbius image $\alpha(s) \approx 0.598 + 0.460i$ has $|\alpha(s)| \approx 0.755 < 1$, well inside the disc; this is consistent with the Cayley transform sending the entire right half-plane $\operatorname{Re} s > 0$ into $\mathbb{D}$. Such points are not zeta zeros (Mathlib's `riemannZeta` has no zeros with $\operatorname{Re} s > 1$), but the disc image is still well-defined.

10 Comparison with the Direct (Non-Singleton) Route

10.1 The direct quotient-class construction

A naive approach to inhabiting `burnolBlaschkeFactor.modelSpaceCarrier` would be to construct a direct quotient class in H^2/BH^2. This requires:

1. A concrete Hardy space H^2 with explicit elements (Vol V's `HardySpaceH2` is opaque);
2. A concrete Hardy submodule BH^2 (Vol V's `blaschkeHardyImage` is also opaque);
3. A concrete quotient construction H^2/BH^2 inhabited by a specific element not in BH^2.

None of these are provided by the Vol V surface. Worse, even if we constructed them in Vol VI, the resulting carrier would be (definitionally) some Vol VI type X, not Vol V's `burnolBlaschkeFactor.modelSpaceCarrier`; we would still need a bridge $X \to$ `burnolBlaschkeFactor.modelSpaceCarrier`, returning us to the same obstruction.

Hence the singleton packet route is preferable: it bottlenecks the opacity exclusively at the bridge, isolating the missing surface.

10.2 The Hardy-RKHS route

Another conceivable route uses the Hardy-RKHS isomorphism $H^2 \simeq \mathrm{RKHS}(K_{\mathrm{Szego}})$ where $K_{\mathrm{Szego}}(z, w) = (1 - z\overline{w})^{-1}$ is the Szegő kernel. The model space K_B then identifies with the RKHS subspace orthogonal to those kernels which factor through B. This route is mathematically cleaner but requires Vol V to expose the RKHS structure on H^2, which it does not. So this route also reduces to the bridge.

10.3 The completion / colimit route

A third route lifts finite-packet model spaces to the canonical Burnol factor by a filtered colimit $K_B \simeq \mathrm{colim}_P\, K_{B_P}$ running over all finite Blaschke packets. This is the route hinted at by RF-05 of the knowledge base. Implementing it requires both:

1. a category structure on finite Blaschke packets (with monomorphism maps $P \hookrightarrow P'$ when P' contains P as a sub-packet);
2. a colimit functor colim : $P \mapsto K_{B_P}$;
3. an isomorphism between this colimit and Vol V's canonical K_B.

Steps (1) and (2) are routine in Mathlib; step (3) is the Vol V bridge in disguise. So this route also bottlenecks on the same obstruction.

10.4 Summary

All three potential routes terminate in the same Vol V opacity. The obstruction module captures this opacity precisely, and the conditional principal theorem absorbs all the surrounding mathematical content into a closed Lean theorem.

11 The Bridge as a Mathematical Statement

11.1 What does it mean to inhabit the bridge?

The bridge `OffCriticalDefectKernelBridge` asks for a function

$$\forall P,\ \texttt{Nonempty}\, P.\texttt{ZeroIndex} \to ((\texttt{finiteBlaschkeModelSpace}\, P).\texttt{carrier} \to \texttt{burnolBlaschkeFactor.}$$

Mathematically, this is the assertion that for any finite Blaschke packet P, there exists *a* map from the Pick coefficient space $P.\texttt{ZeroIndex} \to \mathbb{C}$ to the canonical Burnol model space carrier. Such a map need not be linear, isometric, or canonical; mere existence is enough.

In particular, any one of the following Vol V upgrades would inhabit the bridge:

(a) **Concretization.** Replace `axiom burnolBlaschkeFactor : BurnolBlaschkeFactor` by a `def` producing a concrete factor whose carrier is, say, H^2/BH^2 as a Lean quotient. Then the bridge is the natural inclusion $P.\texttt{ZeroIndex} \to \mathbb{C} \hookrightarrow H^2/BH^2$ sending a coefficient sequence to the corresponding Pick-basis sum modulo BH^2.

(b) **Identification lemma.** Add a Vol V theorem `burnol_modelSpaceCarrier_eq` : `burnolBlaschkeFactor.modelSpaceCarrier` $= (H^2/BH^2)$. Then the bridge is the natural map composed with the equality cast.

(c) **Constructor lemma.** Add a Vol V function `burnol_modelSpaceCarrier_of_disc_zero` : $\forall \alpha \in \mathbb{D},\ B(\alpha) = 0 \to$ `burnolBlaschkeFactor.modelSpaceCarrier` producing the reproducing kernel at any disc zero of B. Combined with a finite-packet selector, this gives the bridge.

11.2 Why option (a) is the intended target

The Vol VI knowledge base RF-01 explicitly identifies (a) as the canonical resolution: "vol6 must define its own concrete replacement for `burnolBlaschkeFactor` (call it `concreteBurnolBlaschkeFactor`) that is definitionally equal or provably equivalent." This is the most invasive but also the most canonical fix; once the carrier is concrete, all three Vol VI papers (03, 04, 05) that depend on it can be closed unconditionally.

11.3 Why we ship the obstruction rather than the fix

Implementing option (a) requires modifying Vol V or shipping a significant Vol VI module that constructs the Burnol/Blaschke factor from scratch in Lean. Both are out of scope for a single sprint paper. Paper 03's role is to discharge *Target 1* of the no-phantom language; the carrier construction is logically prior and should be addressed by a dedicated paper (or by a Vol V upgrade). The obstruction module precisely localises the missing piece, so any future worker can pick it up without re-deriving Paper 03's content.

12 Discussion

12.1 Comparison with the Vol V opaque-axiom semantics

Vol V's choice to declare `burnolBlaschkeFactor` as an axiom reflects the convention that the canonical Burnol/Blaschke factor is treated as data *external* to the Hardy/Mellin axiomatic surface used by the no-phantom route. This keeps the core axiomatic surface lean, but it also pushes the construction of the carrier into Vol VI. The minimal Vol V surface upgrade that would close Paper 03 is to expose any one of:

- a coercion `HardySpaceH2` $\to$ `burnolBlaschkeFactor.modelSpaceCarrier` that factors through the quotient by `blaschkeHardyImage`;
- an equivalence `burnolBlaschkeFactor.modelSpaceCarrier` $\simeq$ `HardySpaceH2/blaschkeHardyIm`
- a constructor $\forall \alpha \in \mathbb{D},\ B(\alpha) = 0 \to$ `burnolBlaschkeFactor.modelSpaceCarrier` producing the reproducing kernel at α.

The third option is the most direct match for our singleton-packet construction; the second is the most categorically natural; the first is the most immediate Hardy-space-theoretic statement.

12.2 Why this paper is not RH-equivalent

Paper 03 *conditionally* discharges Target 1 of the no-phantom language. Combined with Target 2 (Vol VI Paper 06's discharge of `YonedaBlaschkeDetectorCalculus`) and the Vol V master theorem (1), an unconditional Paper 03 closes `RH_classical`. We must therefore explain why our argument is not circular and not RH-equivalent.

The hypothesis `ExistsOffCriticalZero` of Theorem 6.2 is the *negation* of RH; the conclusion is a nonemptiness statement about the canonical Burnol model space. The implication, properly read, is: *any* off-critical zero (whether or not RH holds) creates a nonzero model-space vector. The Vol V master theorem then turns that contradiction into a proof of RH only via combination with the no-phantom predicate `NoPhantomDefect`, which is supplied by Target 2 (Paper 06). Neither Paper 03 nor Paper 06 individually proves RH; their conjunction under the Vol V master rule does. This is the new-language pattern: the mathematical content has been factored into two strictly weaker pieces, each provable by direct construction.

12.3 Limitations

1. **Bridge inhabitation.** Theorem 6.2 is conditional on the bridge. The remaining mathematical work for an unconditional Vol VI closure is the inhabitation of `OffCriticalDefectKernelBridge`, equivalent to discharging RF-01.

2. **Paper 01 carrier choice.** Paper 01's choice of carrier P.`ZeroIndex` $\to \mathbb{C}$ is the simplest valid finite-dimensional carrier; it reproduces the Pick basis up to a noncanonical isomorphism. Other carriers (e.g. a Hardy-space subspace generated by $\{k_{\alpha_i}\}$) would also work and might make the bridge inhabitation easier; the present obstruction is parametric in the carrier and remains valid for any Paper 01 implementation.

3. **Numerical scope.** The Haskell program is a single-point sanity check, not a verification of any specific zeta zero.

13 Conclusion

We have discharged Step (I) and Step (II) of the Paper 03 pipeline as closed Lean theorems

- `singlePointPacket_nonempty` (Theorem 4.3),
- `singlePointModelSpace_nonempty` (Theorem 4.6),
- `finitePacketWitness_of_offcritical` (Theorem 6.3),
- `exists_finitePacket_witness_of_offCritical` (Theorem 6.4),

and we have shipped Step (III) as the conditional principal theorem `off_critical_zero_defect_kernel` `OffCriticalDefectKernelBridge` $\to$ `OffCriticalZeroDefectKernel` together with the typed obstruction module `Vol6.Obstruction.OffCriticalDefectKernelObstruction`. The unconditional principal theorem `off_critical_zero_defect_kernel` is *not* declared in the present module; doing so without a Vol V bridge would violate the Vol VI hard rules.

The Lean development compiles as `lake build Vol6.OffCriticalDefectKernel` with no `sorry`, `admit`, new `axiom`, or new `opaque`. The accompanying Haskell program runs successfully and mirrors Step (I) and Step (II) numerically.

Future work. The next concrete step in the Vol VI pipeline is the production of an inhabitant of `OffCriticalDefectKernelBridge`. This requires either a Vol V upgrade exposing one of the three bridge surfaces listed in Section 5, or a Vol VI replacement of `burnolBlaschkeFactor` by a concrete (non-axiomatic) Burnol factor. The latter is anticipated by RF-01 and would be a substantive mathematical contribution: a Lean-formalised construction of the Burnol/Blaschke factor in terms of fractional-part atoms and finite-rank Hardy quotients.

14 Appendix: Full Lean Module Listing

For completeness we reproduce the full content of `lean/Vol6/OffCriticalDefectKernel.lean` (the principal module) and `lean/Vol6/Obstruction/OffCriticalDefectKernelObstruction.lean` (the obstruction module). These are the canonical Lean artefacts of Paper 03 and should be regarded as part of the paper.

14.1 Principal module

Located at `lean/Vol6/OffCriticalDefectKernel.lean`. The module namespace is `Vol6.OffCriticalDefectKernelNS` and the `noncomputable section` mode is used (because elements are extracted via `Classical.choice` from `Nonempty`).

Key declarations:

- `abbrev alpha (s : Complex) : Complex := mobiusParam s` — disc parameter map.
- `def singlePointPacket (s : Complex) (_h : OffCriticalZero s) : FiniteBlaschkeI` — singleton packet construction.
- `theorem singlePointPacket_nonempty` — closed.
- `def singlePointModelSpace` — singleton model space.
- `theorem singlePointModelSpace_nonempty` — closed via Paper 01.
- `theorem nonempty_burnol_modelSpaceCarrier_of_bridge` — Step (III) under the bridge.
- `theorem off_critical_zero_defect_kernel_of_bridge` — conditional principal theorem.
- `theorem finitePacketWitness_of_offcritical` — audit corollary.
- `theorem exists_finitePacket_witness_of_offCritical` — audit corollary.
- `def off_critical_defect_kernel_status : String` — audit marker.

14.2 Obstruction module

Located at `lean/Vol6/Obstruction/OffCriticalDefectKernelObstruction.lean`. The module namespace is `Vol6.Obstruction`.

Key declarations:

- `noncomputable def mobiusParam (s : Complex) : Complex := (s - 1)/(s + 1)`
- `theorem mobiusParam_one : mobiusParam 1 = 0`
- `def FiniteBlaschkePacketCarrierTransport : Type 1`
- `structure OffCriticalDefectKernelBridge : Type 1`
- `def SingletonBlaschkePacketCarrierTransport : Type 1`

- def singleton_transport_of_full_transport
- def off_critical_defect_kernel_obstruction_status : String

14.3 Build verification

The Lean development is verified by

```
cd lean && PATH="$HOME/.elan/bin:$PATH" \
    lake build Vol6.OffCriticalDefectKernel
```

which terminates with

```
Build completed successfully (2925 jobs).
```

A grep for `sorry|admit|axiom |opaque` in `Vol6/OffCriticalDefectKernel.lean` and `Vol6/Obstruction/OffCriticalDefectKernelObstruction.lean` yields only matches in comments (the words appear in audit prose explaining what is forbidden).

15 Appendix: Full Haskell Driver Listing

The Haskell program lives at `haskell/paper-03-off-critical-defect-kernel/Main.hs`. Key components:

- `mobiusParam :: Complex Double -> Complex Double` — numerical Möbius parameter.
- `isOffCriticalCandidate :: Complex Double -> Bool` — strip and off-line predicate.
- `data SinglePointPacket` — single-zero packet.
- `type SinglePointCarrier = () -> Complex Double` — singleton carrier (mirroring `Fin 1 -> Complex`).
- `singlePointKernelVector :: SinglePointCarrier` — the constant-1 function.
- `kernelVectorNonzero :: Bool` — the nondegeneracy decision.
- `runPipeline :: Complex Double -> IO Bool` — full driver.
- `sanityTable :: IO ()` — multi-point smoke test.

The program runs successfully with

```
runghc haskell/paper-03-off-critical-defect-kernel/Main.hs
```

producing output:

```
Vol6 paper 03 - Off-Critical Defect Kernel (Haskell)
Input s = 0.6 :+ 14.0
  Re s    = 0.6
  Im s    = 14.0
alpha(s) = (s-1)/(s+1) = 0.984 :+ 0.141
  |alpha(s)|  = 0.994
Off-critical candidate? True
Singleton kernel vector v : () -> Complex
  v()      = 1.0 :+ 0.0
  |v|      = 1.0
  nonzero? = True
Assertions:
  [A1] off-critical strip + off line .... True
  [A2] 0 < |alpha(s)| < 1 ............... True
  [A3] zeroPoint() == alpha(s) .......... True
  [A4] kernel vector norm > 0 ........... True
  [A5] kernelVectorNonzero flag ......... True
RESULT: PASS
```

16 Appendix: Vol V Surface Audit

We document the precise Vol V audit performed to certify that no introduction rule for `burnolBlaschkeFactor.modelSpaceCarrier` exists in the no-phantom-language route.

16.1 Files searched

- `Vol5/YonedaNymanTrackA/BurnolCore.lean`
- `Vol5/YonedaNymanTrackA/BurnolFactorization.lean` (excluded from the no-phantom route per knowledge base RF-06; surveyed for completeness only)
- `Vol5/YonedaNymanTrackA/BlaschkeDefectObject.lean`
- `Vol5/YonedaNymanTrackA/HardyModelSpace.lean`
- `Vol5/YonedaNymanTrackA/NoPhantomConcreteAtoms.lean`
- `Vol5/YonedaNymanTrackA/NoPhantomBlaschkeDefect.lean`
- `Vol5/YonedaNymanTrackA/CondensedHilbertDefect.lean`
- `Vol5/YonedaNymanTrackA/RKHSDetectorSemantics.lean`
- `Vol5/YonedaNymanTrackA/FiniteRankRKHSDetector.lean`
- `Vol5/YonedaNymanTrackA/OffCriticalZeroDefect.lean`
- `Vol5/YonedaNymanTrackA/NoPhantomLanguage.lean`
- `Vol5/YonedaNymanTrackA/YonedaCore.lean`
- `Vol5/YonedaNymanTrackA/Basic.lean`

16.2 Search method

For each file, we ran (essentially)

```
grep -n "modelSpaceCarrier" file.lean
```

and inspected each match for context. The matches in `BurnolCore.lean`, `BlaschkeDefectObject.lean`, `NoPhantomConcreteAtoms.lean`, and `NoPhantomLanguage.lean` are uses, not introductions: they reference the carrier via the axiom `burnolBlaschkeFactor.modelSpaceCarrier` and either project from it or assert membership/emptiness predicates. None constructs an inhabitant.

16.3 Conclusion

No Vol V theorem, function, or instance has any of the conclusion shapes (a) identification, (b) injection, or (c) constructor for `burnolBlaschkeFactor.modelSpaceCarrier`, for any concrete type. This certifies that the obstruction module is necessary; without a Vol V upgrade, the unconditional principal theorem cannot be closed by purely axiom-free Vol VI means.

16.4 Implications for the Vol VI sprint

This audit shows that the Vol VI Paper 03 obstruction is not unique: Papers 04, 05, and 06 likely face analogous obstructions for the `DefectDetectedByRationalDilationRepresentable` (RF-02) and `BlaschkeDefectDualizable/Polarized/Regularized` (RF-03) opaque atoms. Each of these papers will need to ship a parallel obstruction module, and the synthesis paper (Paper 07) will need to collect and propagate them. The pattern is uniform: when a Vol V opaque/axiom can be inhabited only externally, Vol VI factors out a typed bridge and delivers conditional theorems. The conditional theorems are then assembled by the orchestrator pipeline; the obstruction propagation tracks exactly what Vol V upgrades remain required for full closure of `RH_classical_new_language`.

Acknowledgements

We thank the orchestrator-pipeline of the Vol VI sprint for parallel delivery of Vol VI Paper 01 (`Vol6.FiniteBlaschkePacket`), without which this paper's Step (II) would have to be re-derived in situ. The Vol V no-phantom-language route, on which all of Vol VI rests, is due to the Vol V development team.

References

[1] J.-F. Burnol, *Sur certains espaces de Hilbert de fonctions entières, liés à la transformation de Fourier et aux fonctions L de Dirichlet et de Riemann*, C. R. Acad. Sci. Paris Sér. I Math., 333 (2001), 273–278.

[2] B. Nyman, *On the One-Dimensional Translation Group and Semi-Group in Certain Function Spaces*, Ph.D. thesis, University of Uppsala, 1950.

[3] A. Beurling, *A closure problem related to the Riemann zeta-function*, Proc. Nat. Acad. Sci. U.S.A., 41 (1955), 312–314.

[4] S. R. Garcia, J. Mashreghi, W. T. Ross, *Introduction to Model Spaces and Their Operators*, Cambridge Studies in Advanced Mathematics, Vol. 148, Cambridge University Press, 2016.

[5] K. Hoffman, *Banach Spaces of Analytic Functions*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[6] N. K. Nikolski, *Operators, Functions, and Systems: An Easy Reading*, Vol. 1: Hardy, Hankel, and Toeplitz, Mathematical Surveys and Monographs, Vol. 92, American Mathematical Society, 2002.

[7] G. Pick, *Über die Beschränkungen analytischer Funktionen*, Math. Ann., 77 (1916), 7–23.

[8] W. Rudin, *Real and Complex Analysis*, 3rd ed., McGraw-Hill, New York, 1987.

[9] E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, 2nd ed., Oxford University Press, 1986.

[10] M. Long, *The No-Phantom Language for Classical RH*, Vol V of the HoTT–Riemann programme, Magneton Research, 2025.

[11] M. Long, *Vol VI Knowledge Base*, Magneton Research, `hott-riemann-hypothesis-vol6/.knowledge-base.md`, 2026.

[12] M. Long, *Concrete Finite Blaschke Packet Model Spaces* (Vol VI Paper 01), Magneton Research, 2026.

[13] M. Long, *Detector Completeness for Finite Blaschke Packets via Gram-Matrix Nondegeneracy* (Vol VI Paper 02), Magneton Research, 2026.

[14] M. Long, *Assembly of the Yoneda/Blaschke Detector Calculus* (Vol VI Paper 06), Magneton Research, 2026.

[15] L. de Moura, S. Ullrich, *The Lean 4 Theorem Prover and Programming Language*, in: A. Platzer, G. Sutcliffe (eds.), Automated Deduction — CADE 28, Lecture Notes in Computer Science, vol. 12699, Springer, Cham, 2021, pp. 625–635.

[16] The Mathlib Community, *The Lean Mathematical Library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), New Orleans, 2020, pp. 367–381.

[17] V. Voevodsky and the Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, Institute for Advanced Study, Princeton, 2013.