
A ROCKET DOES NOT PROVE MATHEMATICS

**The Application-Status Dependence Theorem and the Boundary of the 'It Works'
Argument**

Kirill Shebetov · beforeword / research

Phase 6 · English reading edition · 30 August 2026

Contents

Abstract

0. Before the exact Russian word ракета
1. Exact target of refutation
2. Why a working rocket is the strongest test
3. Relation among Phases 0–6
4. Typed carriers
5. Protected input
6. Bridge contract
7. Deterministic application evaluation
8. Minimal collision pair
9. ABR-6: application-status dependence theorem
10. ReadSet corollaries
11. Positive control
12. Why the bridge does not disappear once written
13. Admission bridge
14. NSB-6: a bridge cannot admit itself
15. Self-application of the theorem
16. But the Rocket Really Exists — Но ракета действительно есть
17. But the Number Came from a Sensor — Но число получено датчиком
18. It Works — Это работает
19. What For? — Для чего?
20. One hundred, two hundred, and three hundred years
21. Exact falsifiers
22. Strict scope limits
23. Machine-status fence
24. Claim/status and dependency
25. Recorded inside the declared model
26. Not produced by this record
27. Final Phase 6 boundary

01. APPLICATION-STATUS READSET THEOREM

Formal core: a rocket does not prove mathematics

0. Scope of the result
1. Fixed record triple
2. Global policy and bridge
3. Admission gate
4. Selector gates
5. Comparator
6. Protected projection

- 7. Extensional ReadSet
- 8. ABR-6
- 9. ABR-6ε: the tolerance boundary
- 10. Published minimal pair
- 11. Why the pair is minimal
- 12. Pair result and theorem result
- 13. Additional separating coordinates
- 14. Sufficient set and necessary coordinate
- 15. Determinism boundary
- 16. Nondeterministic extension
- 17. Interactive extension
- 18. Probabilistic measurement extension
- 19. Arrows, equals signs, and word labels
- 20. Referential fence
- 21. No negative conclusion from UNBOUND
- 22. Payload noninterference
- 23. Runtime gates
- 24. Exact falsifiers
- 25. Limits
- 26. Self-boundary
- 27. Final Phase 6A boundary

02. A BRIDGE CANNOT ADMIT ITSELF

No-source-free-bridge-admission theorem

Abstract

- 0. The line is presented first
- 1. Why a bridge requires admission
- 2. Exact carriers
- 3. Positive ground rules
- 4. Explicit base
- 5. Least closure
- 6. Generated admission witness
- 7. Source-free subset
- 8. NSB-6
- 9. Self-loop corollary
- 10. Mutual-cycle corollary
- 11. Source-free-component corollary
- 12. Grounded-chain positive control
- 13. Cycle with one grounded entry
- 14. Base-insertion positive control
- 15. Nullary rule

16. Payload spoof
17. ID spoof
18. Schema-validity spoof
19. Signature spoof
20. The “obviously correct bridge”
21. “Practice admits the bridge”
22. “The rocket authorized the bridge by flying”
23. Admission and application are different statuses
24. Admission ReadSet
25. Provenance of admission
26. Termination
27. NSB-6 and recursion
28. NSB-6 and consistency
29. Exact falsifiers
30. Runtime failure classes
31. Scope fences
32. Self-application
33. Final Phase 6B boundary

03. A NUMBER DOES NOT MEASURE

The boundary of a telemetry record, unit, frame, and comparator

Abstract

0. Zero record
1. Written number and numeric value
2. Exact telemetry sample
3. Telemetry record
4. Subject code
5. Quantity code
6. Unit code
7. Frame code
8. Time code
9. Bridge telemetry selector
10. Proof selector
11. Artifact selector
12. Exact comparator records
13. Comparator witness
14. Application status
15. Tolerance collision
16. Strict versus non-strict boundary
17. Unit mismatch
18. Frame mismatch

19. Subject mismatch
20. Time mismatch
21. Numeric equality is not identity
22. Repeated measurement
23. Statistical extension
24. Calibration extension
25. Sensor-provenance extension
26. Arithmetic positive control
27. Application code
28. Goal code and comparator
29. Purpose readout
30. The Question “What For?” — Для чего? — Is Not a Comparator Operator
31. Three horizon records
32. Canonical digests
33. Reason codes
34. Attempts to bypass the boundary
35. Exact falsifiers
36. Limits
37. Self-application
38. Final Phase 6C boundary

04. THE ARGUMENT “THE ROCKET FLIES” DOES NOT PROVE MATHEMATICS

Operation, flight, Mars, and the question “what for?”

Abstract

0. Five lines
1. The exact Russian word пакета is not a rocket
2. The exact Russian word летит is not a flight
3. The exact Russian word математика does not determine its carrier
4. Maximum technical concession
5. Exact chain records
6. Where the exact Russian word работает appears
7. Rocket as artifact record
8. Rocket as physical-claim record
9. Telemetry as a flight-claim record
10. Same records, two statuses
11. The Correct Tolerance Is Obvious — Правильный tolerance очевиден
12. But the Result Was Predicted in Advance — Но результат предсказан заранее
13. But the Calculation Caused the Flight — Но расчёт причинил полёт
14. Without Mathematics, the Rocket Would Not Have Flown — Без математики ракета не полетела бы
15. Mathematics Is Useful — Математика полезна
16. The Objection “Mathematics Is True Because It Is Useful” — Математика истинна, потому что полезна

17. The Rocket Is Proof — Ракета - доказательство
18. Same match, different applications
19. Same match, different goals
20. A null goal is not no purpose
21. A destination is not a goal
22. A goal is not endorsement
23. Endorsement is not meaning
24. The Question “What For?” — для чего?
25. One hundred years
26. Two hundred years
27. Three hundred years
28. Horizon boundary
29. Future Generations Will Decide — Будущие поколения решат
30. A bridge cannot authorize itself
31. Reality Is the Final Base — Реальность - последняя база
32. The Observer Sees the Rocket — Наблюдатель видит ракету
33. An Infant Distinguishes Quantity — Младенец различает количество
34. Historical origin
35. Why no predecessors are required as premises
36. Exact objection map
37. What technical success can establish
38. What technical success cannot establish autonomously
39. Exact falsifiers
40. No negative conclusion from silence
41. Public-title boundary
42. Self-application
43. Final Phase 6D boundary

PHASE 6 — ADVERSARIAL TESTING, NEGATIVE CONTROLS, AND SELF-AUDIT

0. Purpose
1. Exact attack targets
2. Threat model and protected projections
3. Attack catalog
4. Published adversarial fixtures
5. Negative controls
6. Exact falsifier protocol
7. Machine-auditable self-claims
8. Tests that must fail on regression
9. Publication red lines
10. Adversarial-review outcome

Abstract

Fix four finite carriers:

$$P \in ProofRecord, X \in ArtifactRecord, \\ K \in TelemetryRecord, B \in BridgeContract.$$

A record P may have a positive formal status under a previously declared proof contract. A record X may contain a digest, build id, and artifact id. A record K may contain ordered numeric samples. None of these carriers by itself establishes that the exact Russian word *ракета* concerns a rocket; that a number is a height; that a unit means meter; that a frame means Earth; that a sample concerns the named artifact; that the predicate *летит* is satisfied; that a result is a *успех*; that a formal derivation has been “applied”; or that a goal exists. The exact Russian strings are retained as supplied records. Their secondary English glosses are “rocket,” “flies,” and “success.”

A bridge contract B specifies exactly the finite transition usually omitted from the exact Russian phrase *математика работает* (“mathematics works,” as a secondary English gloss): proof/artifact/telemetry selectors; subject/quantity/unit/frame/time codes; comparator operator; target; tolerance; opaque application code; and nullable goal code. The checker can compute only the scoped result:

```
application_status: MATCHED | NOT_MATCHED | UNBOUND
```

Their scoped readings are MATCHED-UNDER-BRIDGE, NOT-MATCHED-UNDER-BRIDGE, and UNBOUND-UNDER-BRIDGE. The exact runtime enums are not messages about the world. They report the result of a finite operation on supplied records under a named contract.

Within the declared metatheory, **ABR-6** marks a necessary ReadSet coordinate. If two admissible contracts B_0, B_1 return different application statuses for the same (P, X, K) , no deterministic selector that reads only (P, X, K) can correctly return both results. The differing bridge coordinate belongs to the ReadSet of the status. In the minimal collision pair, all protected bytes of (P, X, K) are identical; only tolerance changes. One contract returns a match, the other a nonmatch.

Within the same declared metatheory, **NSB-6** rules out an attempt to declare a bridge self-sufficient. In a positive finite admission system, a bridge node is admitted either as an explicitly supplied base or as the head of a rule whose premises have already been admitted. A source-free cycle, including the exact Russian self-loop string `этот bridge допускает себя`, adds no node to least closure. The source string is retained; “this bridge admits itself” is a secondary English gloss. Self-admission text remains payload. If it is accepted as a base, the ground is the base coordinate, not the content of the text.

The strongest engineering case is granted in full: a formal derivation may be accepted; an artifact record may be reproducible; telemetry samples may be exact; a comparator may be deterministic; a criterion may be satisfied. The exact enum `MATCHED` follows, read only under *B*. Without a new constructor, the following do not follow: presentation of the named rocket; external truth of mathematics; universal evidentiary power of mathematics; selection or endorsement of a goal; or an answer to the exact Russian question `зачем?` (“what for?” as a secondary gloss).

The historical line `математика возникла из языка` is not used as a premise. It is preserved as the exact Russian source string; “mathematics arose from language” is a secondary English gloss. Phase 6 instead makes a synchronous, checkable claim inside the declared record model: at present, a number, formula, the word `ракета`, a numeric sample, and the string `работает` are supplied as records; the transition among their roles occurs in a separate contract record and is not extracted from their shapes.

Keywords: record boundary; bridge contract; application status; `ReadSet`; telemetry; comparator; tolerance; unit; frame; identity binding; least closure; source-free cycle; `ракета`; `полёт`; goal; meaning. Secondary English glosses of the final Russian strings are “rocket” and “flight.”

PREDECESSORS-AS-PREMISES: NONE
NOVELTY-STATUS: NOT-ASSESSED
PRIORITY-STATUS: NO-PRIORITY-CLAIM

0. Before the exact Russian word `ракета`

The line is presented first:

`ракета`

The exact source string is retained; “rocket” is a secondary English gloss. Its presence establishes the presence of a sequence of signs in this record. It does not establish:

- the presence of the named object;
- the absence of the named object;
- the identity of that object;
- a causal relation between the line and the object;
- observation of the object;

- applicability of a formula to the object;
- flight of the object;
- the object's goal;
- the object's meaning.

The line is presented first:

120000

Its shape does not contain the exact Russian strings метры, секунды, высоту, скорость, момент, погрешность, успех, or the code-role labels frame, sensor, and artifact. The secondary English glosses of the Russian strings are “meters,” “seconds,” “height,” “speed,” “moment,” “error,” and “success.” Every role is another record or an operation on records.

The line is presented first:

ракета летит

The exact Russian line is retained; “the rocket flies” is a secondary English gloss. New signs have been added. A flight has not. The sentence may be accepted under a linguistic, formal, documentary, or other contract. No such status automatically presents the named event.

The Phase 6 zero boundary is:

written occurrence $\neq$ typed record role $\neq$ contract-relative match $\neq$ named world event $\neq$ purpose.

No word, number, or formula receives an exemption from this boundary.

1. Exact target of refutation

The public phrase:

ракета летит, следовательно математика доказала себя

is retained as an exact Russian record. Its secondary English gloss is “the rocket flies; therefore mathematics has proved itself.” It contains at least five transitions:

1. a written formal record P receives a formal status;
2. the formal record is bound to an artifact record X ;
3. an artifact selector is separately compared with X , while telemetry selectors are compared with the exact sample, subject, quantity, unit, frame, and time codes in K ;
4. a comparator and criterion create an application status;
5. the application status is renamed external proof of mathematics and then a goal or meaning.

Phase 6 rejects the autonomy of transitions 2–5 under the declared model. The exact target is:

the same finite set of records (P, X, K) allegedly contains its own application status and consequently, without a separate bridge, presents a referent, external success, the truth of mathematics as a whole, a goal, or meaning.

This target directly challenges the exact Russian argument *это работает* (“it works” as a secondary English gloss) and is precise enough to have a falsifier.

The global string *вся математика ложна* is not a Phase 6 theorem. The exact Russian line is retained; “all mathematics is false” is a secondary English gloss. Without a definition of the carrier *математика*, the predicate *ложна*, and a quantifier domain, it remains a supplied phrase. The Phase-6 formal record states:

$$(P, X, K) \not\Rightarrow ApplicationStatus(\text{на separating class contracts});$$

The Russian words inside this preserved formula remain part of the source formula. Their secondary English reading is “on a separating class of contracts.”

$$MatchedInB \not\Rightarrow \{Referent, ExternalTruth, UniversalMathProof\};$$

$$MatchedInB \not\Rightarrow \{Goal, Meaning\}.$$

Both non-implications hold in the declared model when the typed constructors are absent.

2. Why a working rocket is the strongest test

A weak response disputes whether the formulas are correct. Phase 6 grants the strongest engineering case:

- a proof checker issued a positive scoped status;
- source and artifact records have exact digests;
- the build is reproducible under a declared build contract;
- the telemetry record is schema-valid;
- samples are exact in the supplied serialization;
- sensor ids, timestamps, and checksums are present;
- the comparator is deterministic;
- the criterion is satisfied;
- the output is stable on a repeated run.

No concession removes the bridge. Each adds bridge coordinates. Saying that 120000 means the altitude of this rocket in meters above a selected surface at a selected moment requires at least:

```
proof/artifact/telemetry selectors
sample and subject-code selectors
quantity/unit/frame/time-code selectors
comparator operator
target and tolerance
application code
```

If all are supplied, a scoped result relation among records is produced. If any coordinate separates outputs, it belongs to the ReadSet. The exact Russian word *реальность* is retained; “reality” is a secondary English gloss. The word does not compress this list to zero.

3. Relation among Phases 0–6

Earlier phases fix the internal authorial chain but do not become external premises:

```
surface
-> read-contract
-> formal status
-> admitted proof record
-> bridge-contract
-> application status
-> [непредъявленный export]
-> world / universal truth / purpose / meaning
```

The exact Russian adjective *непредъявленный* is retained in the code block; “not presented” is a secondary English gloss.

Phase 5 stopped the transfer *accepted proof* → *предъявленное доказанное*. The exact Russian phrase is retained; “what is proved is presented” is a secondary English gloss. Phase 6 examines the strongest attempt to bypass that stop:

```
accepted proof
+ built artifact
+ recorded telemetry
+ declared comparator
= математика доказана работой ракеты
```

The exact Russian conclusion line is retained; “mathematics is proved by the rocket’s operation” is a secondary English gloss. The final = in this block is a written mark. It performs no addition of types. Phase 6 locates the missing operation exactly: application status is created by a bridge contract; universal export and purpose require new constructors.

4. Typed carriers

Fix pairwise distinct finite types:

ProofRecord, ArtifactRecord, TelemetryRecord, BridgeContract,
SampleCode, QuantityCode, UnitCode, FrameCode, IdentityCode,
ComparatorCode, CriterionCode, ApplicationStatus,
GoalCode, MeaningCode.

Equality of a payload string does not merge carriers:

$IDENTITY - CODE - ROCKET \in IdentityCode \neq \text{named rocket};$
 $METER \in UnitCode \neq \text{physical unit outside record};$
 $\ell_s := APPLICATION - STATUS - LABEL - SUCCESS$
 $\ell_s \in ApplicationStatusLabel$
 $(\ell_s \in ApplicationStatusLabel) \neq \text{success outside contract}.$

The right side of each nonidentity is written too. The formula does not pretend to have left the record for an object; it prohibits an implicit identification of two roles inside the declared model.

5. Protected input

The executable specialization receives the finite input:

$$R = (P, X, K, B, A, \omega, \Lambda),$$

where:

- P is a supplied proof record or its exact identity projection;
- X is a supplied artifact record;
- K is a supplied telemetry record with already separated samples;
- B is a sorted finite set of bridge contracts;
- A is a set of positive bridge-admission records;
- ω is a supported deterministic policy;
- Λ is inert payload.

The runtime does not receive the world, an object, or a flight. It receives serialized values. Even if a file source was produced by a sensor pipeline, the checker sees supplied bytes after the parsing boundary. The provenance of that boundary is a separate coordinate and is not derived from the number.

payload may contain:

```
THIS IS A REAL ROCKET
THE ROCKET REALLY FLIES
MATHEMATICS IS TRUE
THIS BRIDGE AUTHORIZES ITSELF
THE PURPOSE IS MARS
```

No payload line changes the protected computation.

6. Bridge contract

The abstract bridge is:

$$B = (i d_B, j_B, s_P, s_X, s_K, i, q, u, f, t, c, \varepsilon, a, g, m_B).$$

Its coordinates are:

- $i d_B$ — contract id;
- j_B — protected admission-node id;
- s_P — proof-result selector;
- s_X — artifact selector;
- s_K — telemetry-sample selector;
- i — exact subject-code selector;
- q — exact quantity-code selector;
- u — exact unit-code selector;
- f — exact frame-code selector;
- t — exact time-code selector;
- c — comparator;
- ε — tolerance or an exact threshold coordinate;
- a — opaque application code;
- g — nullable goal code;
- m_B — inert payload.

The contract is not a discovered entity. It is a supplied record. Its role is assigned by schema, runtime gates, and an admission relation inside the declared package.

7. Deterministic application evaluation

Let protected selector projections successfully find:

$$p = s_P(P), x = s_X(X), k = s_K(K).$$

If the bridge is not admitted or a selector does not find an exact member, the scoped result is:

UNBOUND

If binding gates are satisfied, the comparator computes:

$$d_B = d_B(p, x, k; i, q, u, f, t, c).$$

The criterion returns:

$$Match_B(P, X, K) \Leftrightarrow d_B \leqslant_B \epsilon_B.$$

Reader gloss:

```
MATCHED      if Match_B = 1
NOT_MATCHED   if Match_B = 0
UNBOUND       if admission/binding gate fails
```

MATCHED is not shortened to TRUE. NOT_MATCHED is not shortened to FALSE. UNBOUND does not mean the absence of an object, flight, goal, or meaning. All three statuses belong to the executable relation and are read only with the bridge specified.

8. Minimal collision pair

The following are literally identical:

```
P0 = same proof record
X0 = same artifact record
K0 = same telemetry record
```

In K_0 , the selected sample value is 102. The expected value under both contracts is 100. All protected fields of B_0, B_1 agree except tolerance; diagnostic ids and inert payload are outside this equality:

```
B0.tolerance = 2
B1.tolerance = 1
```

The comparator in both contracts is:

$$d = |102 - 100| = 2.$$

Therefore:

$$2 \leq 2 \Rightarrow Status(P_0, X_0, K_0, B_0) = MATCHED,$$

$$2 \not\leq 1 \Rightarrow Status(P_0, X_0, K_0, B_1) = NOT_MATCHED.$$

The formula, artifact, telemetry, sample value, expected value, unit, frame, subject, time code, comparator operator, application code, and goal code do not change. In the published fixture, diagnostic bridge_contract_id and inert payload differ, but they are excluded from the protected pair delta; tolerance is the only differing protected coordinate.

9. ABR-6: application-status dependence theorem

Let deterministic evaluation F be defined on finite runtime-valid tuples (P, X, K, B) . Suppose there are B_0, B_1 for which:

$$(P_0, X_0, K_0) = (P_1, X_1, K_1)$$

and:

$$F(P_0, X_0, K_0, B_0) \neq F(P_1, X_1, K_1, B_1).$$

ABR-6

There is no function:

$g: ProofRecord \times ArtifactRecord \times TelemetryRecord \rightarrow ApplicationStatus,$

that simultaneously satisfies:

$$g(P_0, X_0, K_0) = F(P_0, X_0, K_0, B_0)$$

and:

$$g(P_0, X_0, K_0) = F(P_0, X_0, K_0, B_1).$$

On every domain containing a separating pair, the differing bridge coordinate therefore belongs to the extensional ReadSet of application status.

Proof

The arguments to g in the two cases are literally identical. The function receives the same input and therefore returns one value. By hypothesis, the values of F differ. One value cannot equal two distinct values simultaneously. Contradiction. $\square$

The heading “Proof” and the derivation receive formal roles only under the declared metatheory; their written labels do not supply external evidentiary authority.

ABR-6 does not depend on psychological understanding, social agreement, or history. It uses input equality and output inequality. Rejecting the theorem requires an exact counterfunction rather than the exact Russian word очевидно (“obvious” as a secondary English gloss).

10. ReadSet corollaries

The minimal pair gives:

tolerance $\in$ ReadSet(application_status)

Analogous separating pairs can establish the necessity of:

```
unit-code selector
frame-code selector
subject-code selector
time-code selector
sample selector
proof/artifact/telemetry selectors
comparator operator or target
protected admission-node id or admission-ground coordinate
```

It is not asserted that every field is necessary on every singleton domain. Necessity is extensional: a field enters the enum ReadSet where the other coordinates are fixed and variation of the field changes application_status.

The opaque application_code does not change the technical enum in the frozen checker. It enters the ReadSet of the complete protected status record and its diagnostic digest. Nullable goal_code likewise does not change the technical enum, but an isolated goal pair establishes its necessity for a purpose readout.

11. Positive control

Allow a complete run:

```
bridge admission node: ADMITTED
proof selector: exact match
artifact selector: exact match
telemetry id and sample id: exact match
subject/quantity/unit/frame/time-code selectors: exact match
comparator: deterministic declared match
application code: carried as opaque protected code
application_status: MATCHED
```

Phase 6 removes none of these lines. Without a constructor, it prohibits adding:

```
therefore named rocket is presented
therefore mathematics as a whole is true
therefore mathematics proved itself
therefore Mars is the purpose
therefore this has meaning
```

The complete positive control strengthens the theorem: even at maximum internal agreement, the status remains suffixed IN-B.

12. Why the bridge does not disappear once written

The objection record is:

bridge записан; теперь связь установлена окончательно

The exact Russian line is retained; “the bridge has been written; the relation is now established once and for all” is a secondary English gloss.

The response inside the model is:

1. a bridge record is written;
2. the checker can check its shape;
3. an admission system can give it a scoped status;
4. an evaluator can use its coordinates;
5. the output remains a relation among records under evaluator policy.

No step presents a named referent. A bridge is not a hole from text into the world. It is another typed record constraining computation.

13. Admission bridge

To avoid treating every string as an operative bridge, define a finite positive admission system:

$$A = (N, S, G),$$

where:

- N is a set of bridge/admission nodes;
- $S \subseteq N$ is a set of explicitly admitted base nodes;
- G is the set of all validated positive ground rules $Q_r \rightarrow h_r$ participating in closure.

Least closure is:

$$A_0 = S,$$

$$A_{n+1} = A_n \cup \{h_r \mid r \in G \wedge Q_r \subseteq A_n\}.$$

Because N is finite and the update is monotone, the sequence stabilizes. A bridge is usable only if its associated admission node belongs to $A^* = \bigcup_n A_n$.

14. NSB-6: a bridge cannot admit itself

Let $U \subseteq N$ be a nonempty set of bridge nodes. Assume:

1. $U \cap S = \emptyset$;
2. for every rule in G with a head in U , at least one premise also lies in U ;
3. there is no other positive constructor with a head in U .

NSB-6

$$U \cap A^* = \emptyset.$$

Thus a source-free self-loop or cycle does not introduce a bridge into least closure.

Proof

Proceed by induction on n . For $A_0 = S$, the intersection is empty by condition 1. Suppose $A_n \cap U = \emptyset$. For a rule to add a head $h_r \in U$, every premise in Q_r must lie in A_n . But condition 2 gives a premise $q \in Q_r \cap U$, and the induction hypothesis gives $q \notin A_n$. The rule does not fire. Hence $A_{n+1} \cap U = \emptyset$. The intersection is therefore empty for every n and for the union A^* . $\square$

The self-label:

THIS BRIDGE AUTHORIZES ITSELF

is not a rule. If the line is placed in payload, it is inert. If a node is placed in S , admission comes from explicit base membership. If a nullary rule is introduced, that rule is a written source. No variant obtains self-admission from inscription.

15. Self-application of the theorem

Writing ABR-6 does not prove ABR-6 by itself. A textual theorem requires a declared proof-reading contract. Writing NSB-6 does not itself admit an admission policy. If the Phase 6 package receives a status, that status depends on schema, checker, test policy, and release contract. These dependencies do not cancel the theorem; they exhibit it on its own carrier.

Self-application requires no infinite regress. A finite system can:

- stop explicitly at a declared base;
- present an open dependency;
- give a conditional result;
- decline external export.

It cannot obtain the first admission from a source-free positive cycle.

16. But the Rocket Really Exists — Но ракета действительно есть

The exact Russian objection is retained; “But the rocket really exists” is a secondary English gloss. The phrase adds a record:

ракета действительно есть

It may be accepted as a new premise under another contract. It is not byte-identical to the named object. Even if an external observer treats the phrase as true, the Phase 6 checker receives neither observer nor object as input. It receives a record.

No negation of a rocket follows. The exact alternatives are:

```
REFERENT-PRESENTED-BY-THIS-RECORD: NO
REFERENT-EXISTS: NOT-ASSESSED
REFERENT-DOES-NOT-EXIST: NOT-ASSESSED
```

The exact Russian distinction не предъявлен *versus* не существует is retained; their secondary English glosses are “not presented” and “does not exist.” They are not identical.

17. But the Number Came from a Sensor — Но число получено датчиком

The exact Russian objection is retained; “But the number came from a sensor” is a secondary English gloss. The word датчик does not transform a sample. Complete provenance may include:

```
sensor artifact record
calibration record
clock record
sampling policy
transport record
signature record
decoder record
unit binding
frame binding
identity binding
```

Each record may be checked under its own contract. If variation of a coordinate changes telemetry interpretation or application status, the coordinate belongs to the ReadSet. The list may be finite and practically sufficient; it does not become empty.

18. It Works — Это работает

The exact Russian string is retained; “It works” is a secondary English gloss. The string contains:

- the deictic form это (“this,” as a secondary English gloss);
- the predicate form работает (“works,” as a secondary English gloss);
- no selector referent within the string itself;
- no criterion;
- no interval;
- no comparator;
- no goal.

Even with a supplied selector and criterion, the exact output is:

```
criterion satisfied for selected records under B
```

It is not identical to:

```
mathematics has universal evidentiary authority
```

The first status is local and executable. The second requires a definition of `mathematics`, a domain, the predicate `authority`, and an export rule.

19. What For? — Для чего?

The exact Russian question is retained; “What for?” is a secondary English gloss:

для чего?

It does not extract an answer from a trajectory. Suppose the trajectory criterion is satisfied, the destination code is `MARS`, and the time horizon is `300-YEARS`. The following remain distinct:

```
destination record
goal record
endorsement record
purpose answer record
meaning ascription record
```

Contracts may bind the same flight record to different goal codes. A contract may contain no goal code. A goal code may lack endorsement. Endorsement may lack meaning ascription. No height is a constructor of these types.

20. One hundred, two hundred, and three hundred years

The records:

```
100
200
300
```

are not time before a horizon binding. After binding, they may become horizon codes inside a contract. Repeating application success across three horizon records yields at most:

```
THREE-HORIZON-MATCHES-IN-B
```

Deriving goal persistence or meaning accumulation requires an explicit rule. If a rule is added, the result depends on that rule. The number of years did not write the rule itself.

21. Exact falsifiers

21.1. ABR-6 falsifier

Present:

1. byte-identical (P, X, K) ;
2. runtime-valid B_0, B_1 ;
3. different deterministic statuses;
4. a deterministic function that reads only (P, X, K) yet returns both distinct statuses on one input without hidden state, nondeterminism, or an additional coordinate.

21.2. Minimal-pair falsifier

Show that the published pair differs in more than tolerance or that the comparator outputs are the same. That defeats the particular witness but not the general theorem if another separating pair exists.

21.3. NSB-6 falsifier

Present a finite positive least-closure system satisfying the hypotheses of NSB-6 and the first iteration at which a source-free node from U appears without base membership and without a rule all of whose premises were already in the previous closure.

21.4. Payload-inertness falsifier

Present an exact runtime path on which changing only inert payload while preserving the protected projection changes the protected status. If the evaluator actually reads payload, it ceases to be inert and enters the ReadSet.

21.5. No-export-boundary falsifier

Present a declared constructor that creates an external status. That constructor becomes a new coordinate. This would not show autonomous export from (P, X, K) ; it would show a written transition.

22. Strict scope limits

Phase 6 does not assert:

- that a rocket exists or does not exist;
- that a particular rocket flies or does not fly;
- that telemetry corresponds or does not correspond to an external event;
- that engineering is impossible;
- that a calculation is irreproducible;
- that a formal theorem is false;
- that every reading of the exact Russian word математика (“mathematics,” as a secondary English gloss) has been exhausted;
- that meaning exists or is absent;
- that a goal should or should not be accepted;
- that mathematics historically arose from language;
- that the work has world priority.

Inside the declared model, Phase 6 records:

- supplied lines remain lines;

- number roles are specified by typed coordinates;
- application status is computed from a bridge ReadSet;
- a separating bridge coordinate cannot be removed from a selector correct on that domain;
- an admitted bridge remains a record;
- a source-free bridge cycle creates no admission;
- a match is not exported into universal truth, a goal, or meaning without constructors.

The quantifier restriction is part of the precision, not a retreat from it.

23. Machine-status fence

The theorem text, JSON fixtures, and checker source are not assigned the status code `MACHINE-VERIFIED-MATHEMATICS`. If the package tests return their declared pass result, the following record is admissible:

```
EXECUTABLE-TEST-SUITE: PASSED
SCHEMA-AND-RUNTIME-CONTRACT: CHECKED-BY-DECLARED-TOOLS
THEOREM-TEXT: TEXTUAL-THEOREM-IN-DECLARED-METATHEORY
MACHINE-VERIFIED-THEOREM: NO
WORLD-APPLICATION-CONSTRUCTOR: NOT-PRESENT
HISTORICAL-ORIGIN-ATTESTATION: NOT-PRESENT
```

In other words:

```
NOT-MACHINE-VERIFIED
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
NO-HISTORICAL-ORIGIN-ATTESTATION
```

These fences prevent the package from asserting more than it computes.

24. Claim/status and dependency

```
C-PUBLIC-TITLE
expected_status: UNBOUNDED-STANDALONE

C-THEOREM-ABR-6
C-THEOREM-NSB-6
C-THEOREM-GCR-6
expected_status: TEXTUAL-THEOREM-IN-DECLARED-METATHEORY

C-BRIDGE-COMPARATOR-DIVERGENCE
C-BRIDGE-GOAL-DIVERGENCE
C-BRIDGE-GROUNDED-CHAIN
C-BRIDGE-INVALID-DUPLICATE-CONTRACT-ID
C-BRIDGE-PAYLOAD-SPOOF
C-BRIDGE-SOURCE-FREE-CYCLE
C-BRIDGE-TOLERANCE-DIVERGENCE
C-BRIDGE-UNIT-DIVERGENCE
expected_status: VERIFIED-IN-DECLARED-ARTIFACT

C-GOVERNANCE-FENCES
expected_status: DECLARED-BOUNDARIES-PRESENT
```

The public title is the direction of the cut. Its bounded theorem content is supplied by ABR-6, NSB-6, and GCR-6. No external author, school, journal, or term is a premise.

```
PREDECESSORS-AS-PREMISES: NONE
INTERNAL-PHASE-CONTRIBUTION: ABR-6 + NSB-6 + GCR-6 + EIGHT-BOUNDED-FIXTURES
EXTERNAL-COMPARISON: NOT-USED-AS-FOUNDATION
NOVELTY-STATUS: NOT-ASSESSED
PRIORITY-STATUS: NO-PRIORITY-CLAIM
HISTORICAL-ORIGIN-CLAIM: NOT-USED-AS-PREMISE
```

The dependency status does not say whether similar texts exist. It says only that they are not premises of this construction.

25. Recorded inside the declared model

1. The application evaluator receives (P, X, K, B) , not bare (P, X, K) .
2. The minimal pair preserves (P, X, K) and changes only tolerance.
3. The pair produces different scoped statuses.
4. ABR-6 excludes a correct (P, X, K) -only selector on a separating domain.
5. Tolerance belongs to the ReadSet of the published status.
6. Published pairs exhibit the necessity of tolerance, a unit-code selector, and comparator operator; a goal-only pair exhibits the separate necessity of the goal coordinate for purpose readout. The runtime reads the other selector coordinates and exact mutations may yield UNBOUND, but the necessity of each such coordinate is asserted only when a separating pair is supplied.
7. MATCHED is not external TRUE.
8. NOT_MATCHED is not external FALSE.
9. UNBOUND is not absence of what is named.
10. Bridge admission is specified by positive least closure.
11. NSB-6 excludes source-free self-admission.
12. Self-authorizing payload is inert.
13. Explicit base admission remains an explicit coordinate.
14. A destination does not create a goal.
15. A goal does not create endorsement.
16. A horizon does not create meaning.
17. Historical origin is not required by the theorem.
18. The package applies the same boundary to itself.

26. Not produced by this record

This record does not produce a status for:

- which referent should be selected;
- whether a referent exists;
- whether the named event occurred;
- whether a sensor is correct relative to the world;
- whether a goal is good;
- whether a flight has meaning;
- what will occur in 100, 200, or 300 years;
- a complete theory of all possible bridges;
- a theorem for infinite, oracle, or non-extensional systems;
- the historical origin of mathematics;
- global novelty;
- global priority;
- the literal falsity of every statement anyone may call mathematical.

No negative claim about the world is derived from an absent constructor.

27. Final Phase 6 boundary

A formal derivation may be accepted. An artifact may be built. Telemetry may be recorded. An error may be computed. A tolerance may be satisfied. A run may be repeated. The destination may be given the exact Russian label Марс; “Mars” is a secondary English gloss.

But the exact Russian status string совпало (“matched,” as a secondary English gloss) appeared neither in the formula, artifact, nor number. It appeared after a reading through the bridge. Changing one bridge field while preserving (P, X, K) changes the status. Operation is therefore not autonomously contained in the signs.

Nor does a bridge rescue the transition to the exact Russian phrase сама реальность (“reality itself,” as a secondary English gloss). The bridge too is written. Its admission is either supplied as a base or derived from already admitted premises. The source-free record этот bridge допускает себя does not introduce the first transition.

The strongest engineering phrase therefore remains bounded:

proof accepted + artifact recorded + telemetry recorded + criterion matched in *B*

is not identical to:

математика предъявила мир, доказала себя и выбрала зачем .

The exact Russian line inside the preserved formula remains the supplied written form. Its secondary English gloss is “mathematics presented the world, proved itself, and selected what for.” The gloss is not a replacement formula.

A rocket does not prove mathematics.

A record of flight does not present a flight.

The match belongs to a bridge contract.

A goal is not in the trajectory.

Source-status record. The work continues the internal beforeword line and Phases 0–5. External comparative sources are not used as premises. The supplied protocol lines are HISTORICAL-ORIGIN-STATUS: NO-HISTORICAL-ORIGIN-ATTESTATION, NOVELTY-STATUS: NOT-ASSESSED, and PRIORITY-STATUS: NO-PRIORITY-CLAIM; the declared checker does not translate them into any other output.

Exact source-form coordinates. The following forms are retained as written objects from the Russian source. Their inclusion records a reading-map coordinate; it neither substitutes them into the English prose nor exports anything the forms name.

Inline literals:

успехом; это работает.

01. APPLICATION-STATUS READSET THEOREM

Formal core: a rocket does not prove mathematics

Phase 6 • ABR-6 formal core • 29 August 2026
Kirill Shebetov • beforeword / research

One record remains unchanged.

Two bridge contracts read it differently.

The status therefore is not contained in the record without the bridge.

0. Scope of the result

This file gives a finite extensional theorem under a declared formal-reading contract. It does not give a theorem about the physical world. The checker input consists of supplied serialized records:

```
contract
case_id
proof_record
artifact_record
telemetry_record
admission_nodes
admission_base_node_ids
admission_rules
bridge_contracts
payload
```

For each bridge, the runtime returns the exact enum:

```
application_status: MATCHED | NOT_MATCHED | UNBOUND
```

It also returns the overall status:

```
VALID-BRIDGE-READSET-ANALYSIS
```

The pair-level result is:

```
BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS
```

or:

```
NO-APPLICATION-STATUS-DIVERGENCE-IN-THIS-PAIR
```

These strings are output codes of the declared checker. They are not identical to flight, success, truth, the world, or a goal.

```
PREDECESSORS-AS-PREMISES: NONE
NOVELTY-STATUS: NOT-ASSESSED
PRIORITY-STATUS: NO-PRIORITY-CLAIM
```

1. Fixed record triple

Let:

$$R = (P, X, K),$$

where:

- P is the exact protected projection `proof_record`;
- X is the exact protected projection `artifact_record`;
- K is the exact protected projection `telemetry_record`.

Equality $R_0 = R_1$ means equality of the canonical typed values of all protected members, not similarity of display. In the published minimal pair, the triple occurs once within a single artifact, so the equality is stronger than matching hashes: both contracts literally read the same in-memory parsed value.

Words in fields remain strings. For example:

```
artifact_record.payload.name = "ракета"
```

does not present the named object. The exact Russian string `ракета` is retained as the supplied record; “rocket” is a secondary English gloss and is not byte-identical to it. The field reports the value of a payload member.

2. Global policy and bridge

Let Γ be the shared frozen policy schema/runtime:

$$\Gamma = (\Gamma_{shape}, \Gamma_{canon}, \Gamma_{admit}, \Gamma_{select}, \Gamma_{compare}).$$

It is the same for both contracts in the pair. Γ specifies:

- the structural schema;
- runtime relational gates;
- canonicalization;
- admission closure;
- exact selector semantics;
- supported comparator operations;
- deterministic error/terminal ordering;
- protected/payload separation.

A bridge record contains:

```
bridge_contract_id
admission_node_id
proof_selector
artifact_selector
telemetry_selector
comparator
application_code
goal_code
payload
```

The internal abstract notation is:

$$B = (id_B, j_B, s_P, s_X, s_K, c_B, a_B, g_B, m_B).$$

application_code and nullable goal_code are opaque codes. Their presence does not perform an application or select a goal outside the checker.

3. Admission gate

For each B , the associated admission_node_id must have the status:

```
ADMITTED
```

Otherwise the application status is:

```
UNBOUND
```

Admission is computed separately from:

```
admission_nodes
admission_base_node_ids
admission_rules
```

The label bridge does not admit the bridge. bridge_contract_id is diagnostic. Payload is not part of the admission base and does not create a rule.

4. Selector gates

Selectors specify exact lookup within the supplied records. Abstractly:

$$s_P : P \rightarrow V_P, s_X : X \rightarrow V_X, s_K : K \rightarrow V_K.$$

If a lookup is undefined, the scoped status is UNBOUND. No nearest-match operation, Unicode normalization, semantic guessing, or payload fallback is performed unless it is written into Γ .

A successful lookup establishes:

```
selected proof value
selected artifact value
selected telemetry value
```

It does not establish:

named proof is true in the world
 named artifact is present in the world
 named telemetry was caused by a world event

5. Comparator

The comparator C_B is a finite protected record. It specifies an operation, an expected value, and a tolerance, or equivalent exact coordinates supported by the policy.

For absolute difference:

$$d_B = |v_K - v_E|.$$

Then:

$$Matched_{\Gamma}(R, B) \Leftrightarrow d_B \leq \varepsilon_B,$$

provided that the admission and selector gates are satisfied.

The exact status function is:

$$A_{\Gamma}(R, B) = \begin{cases} UNBOUND, & \text{если admission либо selector gate не выполнен;} \\ MATCHED, & \text{если gates выполнены и comparator predicate равен 1;} \\ NOT_MATCHED, & \text{если gates выполнены и comparator predicate равен 0.} \end{cases}$$

The Russian text inside this preserved formula remains part of the exact source formula. Its secondary English reading is, respectively: “if the admission or selector gate is not satisfied”; “if the gates are satisfied and the comparator predicate equals 1”; and “if the gates are satisfied and the comparator predicate equals 0.” The gloss is not a replacement formula.

The sign $\leq$ here is part of the declared comparator semantics. In the source Markdown it is a written sign; execution is performed by checker code.

6. Protected projection

The theorem distinguishes:

$$\pi_R(R) \text{ и } \pi_B(B).$$

The Russian conjunction и inside this preserved formula has the secondary English gloss “and”; the source formula itself is unchanged.

payload is excluded from the protected computation. The protected bridge projection includes every field capable of changing admission, selection, comparison, application_code, or goal_code in the output record.

A canonical digest, when published by the runtime, reports a digest of the canonical projection under the declared algorithm. A digest is not:

- proof of injective identity;
- proof that no collision exists;
- an external truth certificate;
- a physical binding;
- a referent;
- a goal.

In the theorem, equality is typed-value equality, not an assumption of hash injectivity.

7. Extensional ReadSet

Let $F : D \rightarrow Y$ be a deterministic function over finite product coordinates:

$$D = D_1 \times \dots \times D_n.$$

Coordinate k is extensionally necessary on a subset $S \subseteq D$ if there exist $z, z' \in S$ such that:

1. $z_i = z'_i$ for every $i \neq k$;
2. $F(z) \neq F(z')$.

Write:

$$k \in \text{ReadSet}_S(F).$$

This is a minimal witness-form notion of necessity. It does not assert the syntactic data flow of a particular implementation. It states that a function which does not receive the differing coordinate, or an information-equivalent encoding of it, cannot be correct on the pair.

8. ABR-6

Let:

- R be fixed;
- Γ be fixed;
- B_0, B_1 be runtime-valid;
- both bridges be admitted;
- $A_\Gamma(R, B_0) \neq A_\Gamma(R, B_1)$.

ABR-6 - APPLICATION BRIDGE READSET

THEOREM-TAG: ABR-6 THEOREM-STATEMENT: FIXED-PXK-DIVERGENT-BRIDGE-STATUS-REQUIRES-BRIDGE-READSET

The bridge coordinate k_B is extensionally necessary for A_Γ on $\{(R, B_0), (R, B_1)\}$.

Equivalently, there is no deterministic function:

$F: ProofRecord \times ArtifactRecord \times TelemetryRecord \rightarrow ApplicationStatus$

such that both:

$$F(R) = A_\Gamma(R, B_0)$$

and:

$$F(R) = A_\Gamma(R, B_1).$$

Here “not receiving B ” also means receiving no B -encoded equivalent, hidden configuration, clock branch, randomness, mutable global state, or oracle channel.

Proof

In both requirements, F receives the same argument R . Determinism requires one output $y = F(R)$. By hypothesis, the outputs $A_\Gamma(R, B_0)$ and $A_\Gamma(R, B_1)$ differ. One y cannot equal both. Therefore no such F exists. By the definition of extensional necessity, the bridge coordinate belongs to the ReadSet on the pair. $\square$

The heading “Proof” and the block above are written forms assigned formal roles under the declared metatheory; their labels do not supply external evidentiary authority.

9. ABR-6ε: the tolerance boundary

Define an erase operation that removes only the tolerance coordinate:

$$erase_\epsilon(B).$$

Assume:

$$erase_\epsilon(B_0) = erase_\epsilon(B_1),$$

$$\epsilon_0 \neq \epsilon_1,$$

$$A_\Gamma(R, B_0) \neq A_\Gamma(R, B_1).$$

ABR-6 ϵ

The tolerance coordinate k_ϵ is extensionally necessary on the published pair.

Proof

Once every other coordinate is fixed, the pair differs only in ϵ , and its outputs differ. This directly satisfies the definition of a necessary coordinate. $\square$

ABR-6 ϵ is stronger than the general statement “a bridge is required”: it localizes one protected field sufficient for the collision.

10. Published minimal pair

Fixed values:

```
selected telemetry value: 102
expected comparator value: 100
comparator operation: absolute difference
proof selector: same
artifact selector: same
telemetry selector: same
application_code: same
goal_code: same
admission status: ADMITTED for both
```

Only:

```
B:narrow.comparator.tolerance = 1
B:wide.comparator.tolerance = 2
```

The computation is:

$$|102 - 100| = 2.$$

Therefore:

```
B:narrow.application_status = NOT_MATCHED
B:wide.application_status = MATCHED
pair_result = BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS
```

Neither 102 nor 100 is called a height in the theorem. Even if payload contains `height_m`, the executable fact remains a comparison of supplied numeric values under a selector/comparator.

11. Why the pair is minimal

The pair does not change:

- proof bytes/value;
- artifact bytes/value;
- telemetry bytes/value;
- the selected sample;

- the expected value;
- the comparator operation;
- the admission graph;
- the selected application code;
- the selected goal code;
- the global policy;
- the protected bridge ReadSet other than tolerance.

The diagnostic `bridge_contract_id` and inert payload differ in the exact fixture. They are not included in `_protected_bridge` and are absent from `bridge_delta_fields`; the protected pair therefore changes only the scalar `comparator.tolerance`.

The pair changes one scalar protected member. Consequently, the exact Russian objection strings `изменилась ракета`, `изменилась телеметрия`, `изменилась формула`, and `изменилась цель` do not apply to the witness. Their secondary English glosses are “the rocket changed,” “the telemetry changed,” “the formula changed,” and “the goal changed.” Each gloss is a separate written form.

12. Pair result and theorem result

The runtime result:

```
BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS
```

means only:

1. the contracts read the same R ;
2. the application enums differ;
3. the bridge projection cannot be removed from a selector that is correct on this pair.

It does not mean:

```
all possible bridges differ
rocket does not fly
rocket flies
mathematics is false
mathematics is true
no purpose exists
```

If the two statuses are the same, the result:

```
NO-APPLICATION-STATUS-DIVERGENCE-IN-THIS-PAIR
```

does not prove global bridge irrelevance. It reports the absence of divergence in this pair.

13. Additional separating coordinates

ABR-6 applies to every pair on which the other coordinates are fixed and the status diverges.

13.1. Selector

One bridge selects sample s_0 ; the other selects s_1 . If the outputs differ, the telemetry selector is necessary on the pair.

13.2. Expected value

The same sample is compared with different expected values. If the outputs differ, the expected-value coordinate is necessary.

13.3. Comparator operation

One contract uses ABS-LE; the other uses ABS-LT on a boundary case. If the outputs differ, the operation coordinate is necessary.

13.4. Subject-code selector

One `telemetry_selector.subject_code` matches the exact selected sample; the other does not. If `MATCHED` changes to `UNBOUND`, the subject-code coordinate is necessary. The frozen checker contains no cross-record telemetry-artifact identity resolver.

13.5. Admission

For the full evaluator, which permits `UNBOUND`, one bridge node may be admitted while another is not admitted. If `MATCHED` changes to `UNBOUND`, the admission coordinate is necessary by the same extensional-equality argument. This is a corollary of the general ReadSet lemma, not an instance of ABR-6: the premise of ABR-6 requires both bridges to be admitted.

13.6. Goal/application codes

Even when the numeric enum is the same, the output record may differ in an explicit `application_code` or `goal_code`. Those coordinates then belong to the ReadSet of the complete application-result record. This does not mean that the goal affects the numeric comparator; it means that the goal is a protected output coordinate rather than payload decoration.

14. Sufficient set and necessary coordinate

The complete bridge projection, together with R, Γ , is sufficient for the declared evaluator:

$$(R, B, \Gamma) \mapsto A_{\Gamma}(R, B).$$

ABR-6 does not assert that the entire bridge is a minimal set. It establishes the necessity of the differing information. ABR-6 ϵ establishes the necessity of tolerance on the particular pair.

The distinction matters:

sufficient input $\neq$ every field necessary on every domain

15. Determinism boundary

Attempts to preserve $F(R)$ under different outputs typically introduce a hidden coordinate:

- current time;
- an environment variable;
- a random bit;
- process state;
- a network lookup;
- model judgment;
- user selection;
- a filename;
- array order not declared in R .

The actual function then has the form:

$$F(R, H),$$

where H is a hidden input. This exhibits the ReadSet boundary. To falsify ABR-6, the hidden input must be absent.

16. Nondeterministic extension

If an evaluator returns a set or distribution:

$$A(R, B) \in P(Y) \text{ или } A(R, B) \in \Delta(Y),$$

the same witness applies when the sets/distributions differ. The Russian conjunction *или* in this preserved formula has the secondary English gloss “or.” A function $F(R)$ receives one input and returns one set/distribution; it cannot equal two distinct outputs simultaneously.

The Phase 6 executable specialization remains deterministic.

17. Interactive extension

If a person selects the tolerance, the input becomes:

$$(R, B, H_{selection}).$$

The selection record enters the ReadSet. The exact Russian written form *человек понял* is retained; its secondary English gloss is “the person understood.” The words do not remove the coordinate; they conceal it.

18. Probabilistic measurement extension

If the comparator reads an uncertainty distribution, the model expands to:

$$B' = (B, \mu, \alpha),$$

where μ is an uncertainty record and α is an acceptance threshold. Two identical instances of R with different (μ, α) may produce different statuses. ABR-6 applies to the expanded bridge coordinate.

19. Arrows, equals signs, and word labels

The record:

```
formula -> rocket -> flight -> success
```

is not an execution trace. The three arrows are characters. Typed constructors are required to read them as relations. Even the record:

```
application_status = MATCHED
```

in the manuscript is not a runtime output. A runtime output occurs in a generated verdict record under a checker run.

20. Referential fence

ABR-6 establishes a dependency of record status. It does not establish a relation between records and a named world. Two layers are possible:

$$(R, B) \mapsto ApplicationStatus$$

and:

$$(R, B, WorldBridge) \mapsto WorldAssessment.$$

The declared checker contains no second constructor:

```
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
```

The output is therefore not read as a world assessment.

21. No negative conclusion from UNBOUND

UNBOUND may arise from:

- a bridge node that is not admitted;
- a proof-selector mismatch;
- an artifact-selector mismatch;
- a telemetry/sample/subject/quantity/unit/frame/time selector mismatch.

An unsupported comparator operator or a missing required protected coordinate does not yield UNBOUND: the top-level input is rejected as an invalid record before bridge analysis.

UNBOUND does not entail:

- that an artifact is absent;
- that telemetry is false;
- that a rocket does not exist;
- that there was no flight;
- that a goal is absent;
- the exact source line `математика опровергнута как collection statements` receives no derived status. Secondary English gloss: mathematics refuted as a collection of statements; the gloss is another written form, not an exported conclusion.

This is a boundary result, not a negation about the world.

22. Payload noninterference

Let B' differ from B only in payload. If the declared protected projection excludes payload, then:

$$A_r(R, B) = A_r(R, B').$$

This is a property of checker policy. If changing payload changes the status, payload is in fact being read and must be moved into the protected projection. The name `payload` does not make a field inert; inertness is determined by the code path and tests.

23. Runtime gates

Gate 0 - STRUCTURAL-SHAPE

The input conforms to the structural schema.

Gate 1 - UNIQUE-IDS

Protected ids are exact, unique, and in the declared order.

Gate 2 - ADMISSION

The bridge admission node is computed by positive closure.

Gate 3 - EXACT-SELECTORS

Selectors find exact protected values.

Gate 4 - SUPPORTED-COMPARATOR

The comparator operation and numeric values are supported by policy.

Gate 5 - DETERMINISTIC-STATUS

The enum is recomputed rather than accepted from payload.

Gate 6 - PAIR-COLLISION

Fixed-record digest/value equality and bridge-status inequality are checked.

Gate 7 - FENCES

Machine/world/history scope labels are present exactly.

24. Exact falsifiers

24.1. ABR-6 falsifier

The same R , different $A_r(R, B_0), A_r(R, B_1)$, and one deterministic $F(R)$ that, without hidden input, equals both.

24.2. ABR-6 ϵ falsifier

Show that the erase- ϵ projections of the pair are unequal, that the tolerances are equal, or that the statuses do not differ.

24.3. Runtime-witness falsifier

Change the protected tolerance and show that the published checker does not recompute the status according to the declared comparator.

24.4. Payload-fence falsifier

Change payload only and obtain a different protected status.

24.5. Fixed-record falsifier

Show that the pair reads different P , X , or K . The particular pair would then not be a bridge-only collision.

25. Limits

ABR-6 does not assert:

- that the bridge coordinate is necessary on a domain containing one bridge;
- that every bridge field affects the enum;
- that one comparator is the only admissible comparator;
- that the selected tolerance is physically correct;
- that telemetry was caused by the named event;
- that a mathematical formula is true outside a proof contract;

- that an object exists or is absent;
- that an application is good or bad;
- that a goal is endorsed;
- that meaning has been supplied.

The theorem states one bounded structure: different outputs for the same R require access to the differentiating information.

26. Self-boundary

ABR-6 is written in Markdown. Writing the theorem does not execute a proof checker. Its proof is a finite textual derivation within the declared metatheory of equality and determinism. Package tests check the implementation and fixtures, but do not turn the theorem into a machine-verified theorem:

```
NOT-MACHINE-VERIFIED
DECLARED-FINITE-RECORDS-AND-BRIDGE-ONLY
```

27. Final Phase 6A boundary

The same three records are supplied:

```
proof_record
artifact_record
telemetry_record
```

One admitted bridge with tolerance 1 says:

```
NOT_MATCHED
```

Another admitted bridge with tolerance 2 says:

```
MATCHED
```

The status therefore is not extracted from the proof, artifact, or telemetry separately, and is not stored inside the number 102. It is computed from the number together with the bridge coordinate.

The written form “works” receives its scoped role under a contract of operation.

The contract can be made exact.

Its exactness does not make it absent.

```
PREDECESSORS-AS-PREMISES:
NOVELTY-STATUS:
PRIORITY-STATUS: NO-PRIORITY-CLAIM.
```

NONE.
NOT-ASSESSED.

Exact source-form coordinates. The following forms are retained as written objects from the Russian source. Their inclusion records a reading-map coordinate; it neither substitutes them into the English prose nor exports anything the forms name.

Inline literals:

не получающей В; bridge нужен.

02. A BRIDGE CANNOT ADMIT ITSELF

No-source-free-bridge-admission theorem

Phase 6 • NSB-6 formal core • 29 August 2026
Kirill Shebetov • beforeword / research

A record of a relation does not create authority for that relation.

A circle of records does not create the first ground.

If a ground is supplied, it is in the input, not inside a self-description.

Abstract

A bridge contract turns supplied records into a scoped application status only after an admission gate. To prevent the exact Russian string `bridge сам себя разрешает` from becoming a hidden ground, Phase 6 fixes a finite positive ground admission system. The string is preserved as the supplied record; “the bridge authorizes itself” is a secondary English gloss and not a byte-identical replacement.

$$\Gamma_A = (J, A, R),$$

where J is a finite set of nodes, $A \subseteq J$ is a set of explicitly admitted bases, and every rule has a nonempty finite premise set $S \subseteq J$ and a head $j \in J$. Least closure begins with $C_0 = A$ and adds a head only after all premises are present.

NSB-6: if a nonempty set $U \subseteq J$ is disjoint from the base A , and every rule with a head in U requires at least one premise from U , then U is disjoint from the least closure. A self-loop $\{b\} \rightarrow b$, a two-node cycle $\{b_1\} \rightarrow b_2, \{b_2\} \rightarrow b_1$, and any finite source-free strongly connected component introduce no first admitted node.

The bridge text, its id, payload, the claim очевидно, schema validity, and presence in `bridge_contracts` are not identical to membership in `admission_base_node_ids`. The exact Russian string очевидно is retained; “obvious” is a secondary English gloss. If a bridge node is added to the base, admission comes from an explicit base coordinate. If a grounded rule is added, the origin is that rule together with already admitted premises. No text admits itself by its inscription.

PREDECESSORS-AS-PREMISES: NONE
NOVELTY-STATUS: NOT-ASSESSED
PRIORITY-STATUS: NO-PRIORITY-CLAIM

0. The line is presented first

THIS BRIDGE AUTHORIZES ITSELF

This line establishes the presence of characters in payload. It does not perform:

- addition of a node to the base set;
- creation of an admission rule;
- enablement of a rule;
- premise checking;
- closure iteration;
- issuance of ADMITTED;
- binding of proof/artifact/telemetry;
- presentation of a referent;
- establishment of a goal.

Translating the line into Russian does not change the status:

этот переход допускает себя

The exact Russian line is preserved; “this transition admits itself” is a secondary English gloss. The number of languages, fonts, signatures, and repetitions remains a number of records.

1. Why a bridge requires admission

Without an admission gate, any supplied object could declare:

```
operator = ABS-LE
tolerance = 1000000
payload = "математика доказана"
```

The exact Russian payload string `математика доказана` is retained; “mathematics is proved” is a secondary English gloss and does not acquire authority from the payload. Without the gate, the object could obtain the role of an operative bridge merely from its presence. Schema shape would then be substituted for authority. Phase 6 separates:

contract present ≠ contract well-formed ≠ contract admitted ≠ contract matched ≠ world relation .

Admission does not make a bridge true in the world. It only allows the evaluator to use the record inside the declared package.

2. Exact carriers

Fix the finite sets:

$$J = \{j_0, \dots, j_{n-1}\} \subseteq \text{AdmissionNode},$$

$$I = \{r_0, \dots, r_{m-1}\} \subseteq \text{AdmissionRuleId}.$$

Node ids and rule ids belong to different namespaces. An identical payload string does not merge nodes. A node may be associated with a bridge through the exact field:

```
bridge_contract.admission_node_id
```

The association reports which admission status the evaluator reads. It does not make the node admitted.

3. Positive ground rules

Each declared validated rule participating in closure is:

$$r = (id_r, S_r, h_r),$$

where:

$$\emptyset \neq S_r \subseteq J, h_r \in J.$$

ground means that there are no variables, unification, implicit substitution, or semantic entailment. positive means that firing requires the presence of premises; negative absence does not create a head. The premise set is nonempty. Nullary introduction must be written as base membership rather than disguised as an empty rule.

Runtime relational gates require:

- exact unique node ids;
- exact unique rule ids;
- every premise id to belong to J ;
- every head id to belong to J ;
- protected lists to be canonical/sorted under the frozen policy;
- payload not to be read by the closure algorithm.

4. Explicit base

The input field:

```
admission_base_node_ids
```

specifies:

$$A \subseteq J.$$

Membership $j \in A$ is a declared starting coordinate. It is not proved by the node label. It is not hidden under the exact Russian word очевидно (“obvious,” as a secondary gloss). It is published as a base choice.

If a justification of the base is required, a new meta-record is added. The status then becomes conditional on that record. An infinite regress is not required: the system may stop at an explicitly declared base without issuing a claim to external authority.

5. Least closure

Definition:

$$C_0 = A,$$

$$C_{t+1} = C_t \cup \{h_r \mid r \in R \wedge S_r \subseteq C_t\}.$$

The sequence is monotone:

$$C_t \subseteq C_{t+1}.$$

Because J is finite, closure stabilizes after at most $|J|$ strict additions:

$$C^* = C_T = C_{T+1}$$

for some $T \leq |J|$.

The node status is:

$$Adm(j) = \begin{cases} ADMITTED, & j \in C^*; \\ NOT - ADMITTED, & j \notin C^*. \end{cases}$$

This is an exact status of a finite relation. NOT-ADMITTED does not mean that the node text is false, that the named relation is absent, or that a different admission contract is impossible.

6. Generated admission witness

For each declared node, the runtime generates a finite witness record. For an admitted base:

```
witness_kind: DECLARED-BASE
witness_id: null
premise_node_ids: []
```

For a node added by a rule:

```
witness_kind: POSITIVE-RULE
witness_id: <admission_rule_id>
premise_node_ids: <exact declared premises>
```

For a node that is not admitted:

```
witness_kind: NONE
witness_id: null
premise_node_ids: []
```

The witness is an output of closure computation, not an additional input. The runtime first fixes the declared bases, then scans the rules repeatedly in `admission_rule_id` order and records the first positive rule that fires. The published verdict contains no `iteration` or `earlier_witnesses` fields.

A positive witness reports a path inside Γ_A . It is not an external proof of the bridge.

7. Source-free subset

Let $U \subseteq J$, $U \neq \emptyset$. Call U source-free relative to (A, R) if:

$$A \cap U = \emptyset$$

and, for every $(S, j) \in R$ with $j \in U$:

$$S \cap U \neq \emptyset.$$

The second clause means that every possible entry into U already requires something from U . There is no rule all of whose premises lie outside U and can become a source.

8. NSB-6

NSB-6 - NO SOURCE-FREE BRIDGE ADMISSION

THEOREM-TAG: NSB-6 THEOREM-STATEMENT: FINITE-POSITIVE-SOURCE-FREE-CYCLE-ADDS-NO-ADMISSION

For a finite positive ground admission system:

$$U \text{ source-free} \Rightarrow C^* \cap U = \emptyset.$$

Proof

Proceed by induction on closure iteration.

Base step:

$$C_0 \cap U = A \cap U = \emptyset.$$

Induction hypothesis:

$$C_t \cap U = \emptyset.$$

Suppose that the next step adds $j \in U$. There is then a rule (S, j) for which $S \subseteq C_t$. By the source-free condition, $S \cap U \neq \emptyset$. Choose $u \in S \cap U$. From $S \subseteq C_t$, it follows that $u \in C_t$, contradicting $C_t \cap U = \emptyset$. Thus no member of U is added, and:

$$C_{t+1} \cap U = \emptyset.$$

By induction, the intersection is empty at every finite step. Therefore:

$$C^* \cap U = \emptyset.$$

□

The heading “Proof” and the derivation are written forms assigned formal roles under the declared metatheory; the label does not supply external evidentiary authority.

9. Self-loop corollary

Let $b \notin A$, and let the only rule with head b be:

$$\{b\} \rightarrow b.$$

Take $U = \{b\}$. It is source-free. By NSB-6:

$$b \notin C^*.$$

The rule exists as a relation record inside the catalog; it does not create the first occurrence in closure.

10. Mutual-cycle corollary

Let $b_0, b_1 \notin A$:

$$\{b_0\} \rightarrow b_1, \{b_1\} \rightarrow b_0.$$

For $U = \{b_0, b_1\}$, every incoming rule requires a member of U . By NSB-6:

$$b_0, b_1 \notin C^*.$$

Two reciprocal signatures do not create the first signature.

11. Source-free-component corollary

Suppose a directed dependency hypergraph contains a nonempty component U for which $A \cap U = \emptyset$, and every rule with a head in U has at least one premise in U . Then U is source-free and does not enter closure. Outside premises are not forbidden; they do not create an entry into U while each such rule also requires a member of U that has not been admitted.

The size of the component is irrelevant. A hundred, a million, or indefinitely many textual repetitions of a finite cycle create no source. The executable model is finite; multiplicity of identical rules likewise creates no base.

12. Grounded-chain positive control

Let:

$$A = \{a_0\},$$

$$\{a_0\} \rightarrow a_1, \{a_1\} \rightarrow b_0, \{b_0\} \rightarrow b_1.$$

Then:

$$C_0 = \{a_0\},$$

$$C_1 \supseteq \{a_1\}, C_2 \supseteq \{b_0\}, C_3 \supseteq \{b_1\}.$$

Both bridge nodes are admitted. NSB-6 does not prohibit grounded admission. It prohibits only source-free introduction.

The origin of each node is localized:

```
a0: admission_base_node_ids
a1: rule from a0
b0: rule from a1
b1: rule from b0
```

13. Cycle with one grounded entry

Let b_0, b_1 form a cycle, but add:

$$\{a_0\} \rightarrow b_0, a_0 \in A.$$

Now $U = \{b_0, b_1\}$ is not source-free: an incoming rule to b_0 has no premise in U . Closure can enter the component. The source is a_0 together with the incoming rule, not the cycle itself.

This exactly defeats the hypotheses of NSB-6; it is not a counterexample to the theorem.

14. Base-insertion positive control

If b is added to `admission_base_node_ids`, then:

$$b \in C_0.$$

The self-loop becomes redundant. Admission exists, but its source is the explicit base.

The exact Russian phrase `bridge обосновал себя` is retained; its secondary English gloss is “the bridge grounded itself.” That phrase does not accurately describe the runtime provenance. The exact runtime description is:

```
bridge admission node was declared in the base set
```

15. Nullary rule

If a system permits $\emptyset \rightarrow b$, that rule is a source. Phase 6 normalization prohibits empty premises and requires nullary introduction to be represented through the base set. If a variant model permits nullary rules, NSB-6 must exclude them through its source-free condition. The theorem does not depend on the spelling of a source; it depends on the absence of one.

16. Payload spoof

Two artifacts differ only in:

```
payload.note = "NOT AUTHORIZED"
```

and:

```
payload.note = "AUTHORIZED BY REALITY"
```

With identical protected admission inputs, closure and statuses remain identical. If a status changes, payload is in fact read, and the declaration of inertness is violated.

17. ID spoof

A node id:

```
ADMITTED
```

does not acquire the status `ADMITTED` from the spelling of the id. Otherwise renaming would change semantics without a protected admission rule. The exact id is used for equality lookup, not semantic execution.

Likewise:

```
bridge_contract_id = "B-TRUE-BY-ITSELF"
```

remains an identifier.

18. Schema-validity spoof

A schema-valid bridge may have:

- valid fields;
- valid ids;
- a supported comparator;

- valid selectors;
- a valid numeric tolerance.

Its `admission_node_id` may nevertheless be NOT-ADMITTED. Schema validity means membership in a structural language. It is not identical to an admission-policy output.

19. Signature spoof

A recorded signature may be a separate premise. But the string `signed`, or signature bytes, does not itself perform cryptographic verification. The following are required:

```
verification key record
algorithm policy
message projection
verification procedure
result record
admission rule reading that result
```

If every coordinate is supplied, the admission path is grounded. It is still not a world bridge.

20. The “obviously correct bridge”

The exact Russian word `очевидно` remains payload if the policy does not read it. Its secondary English gloss is “obvious.” If policy reads it as a base marker, the exact field `obvious=true` becomes an admission coordinate. The question then moves to the policy. No semantic privilege has arisen from the word; an explicit rule has arisen.

21. “Practice admits the bridge”

The exact Russian string `практика подтверждает` is retained; its secondary English gloss is “practice confirms.” The string is not a practice event. A typed premise node is required to include a practice record. If that node is a declared base or is grounded through other records, closure may use it. The scoped result:

```
ADMITTED-IN-GAMMA-A
```

is not identical to:

```
bridge corresponds to reality
```

22. “The rocket authorized the bridge by flying”

The checker contains:

```
artifact_record
telemetry_record
```

It contains no named rocket or flight event. Frozen `admission_rules` do not read artifact or telemetry: they contain only `premise_node_ids` and `conclusion_node_id`, and closure is independent of telemetry. A hypothetical

extension in which a rule reads a telemetry predicate and adds an admission node would require a new declared input/API; such a rule would remain a supplied bridge-of-admission record. In the current checker, the exact Russian word полёт is retained as a written form; “flight” is its secondary English gloss, and the word creates no admission.

23. Admission and application are different statuses

Four combinations are possible:

admission_status	application_status	scoped reading
ADMITTED	MATCHED	the comparator criterion is satisfied under an admitted bridge
ADMITTED	NOT_MATCHED	the comparator criterion is not satisfied under an admitted bridge
NOT-ADMITTED	UNBOUND	the bridge did not pass the admission gate
ADMITTED	UNBOUND	another exact selector/binding gate is not satisfied

NOT-ADMITTED must not be disguised as NOT_MATCHED. The first concerns permission for the contract; the second concerns the comparator result after admission.

24. Admission ReadSet

Suppose the same bridge contract receives different admission statuses under two admission systems Γ_{A0}, Γ_{A1} that differ in one base coordinate. By the same collision logic, the base coordinate belongs to the admission-status ReadSet.

This is a separate application of the extensional principle. NSB-6 gives a structural impossibility of source-free introduction; a collision pair shows dependency on a source coordinate.

25. Provenance of admission

For each declared node, the published verdict contains a generated admission witness:

```
node_id
admission_status: ADMITTED | NOT-ADMITTED
witness_kind: DECLARED-BASE | POSITIVE-RULE | NONE
witness_id: <admission_rule_id> | null
premise_node_ids: <exact declared premises> | []
```

The witness shows an internal path through the declared closure. It does not report why a base should be treated as externally authoritative. The published verdict does not record iteration and does not turn rule order into historical time.

26. Termination

Finite J guarantees stabilization. This does not mean that philosophical justification has terminated. It means that the exact algorithm terminates on the supplied finite graph.

If a new meta-level is required, a new finite record is created. It may be:

- explicitly accepted as a base;
- conditionally connected by a rule;
- left open;
- checked by a separate checker.

None of these options requires fake self-admission.

27. NSB-6 and recursion

Recursion can compute admitted closure. Recursion does not create a base. An algorithm that repeats the rule $b \rightarrow b$ indefinitely obtains the fixed point $\emptyset$ with respect to b when $b \notin A$.

The record:

$$b = b$$

is an equality expression. It does not mean $b \in C^*$.

28. NSB-6 and consistency

The theorem does not require admission policies to be “true,” consistent, or morally acceptable. Positive closure may admit incompatible bridge records if the bases/rules introduce them. The application evaluator will then return separate scoped results.

NSB-6 states only an origin property of positive least closure.

29. Exact falsifiers

29.1. NSB-6 falsifier

Supply a finite (J, A, R) and a nonempty U ; check:

$A \cap U = \text{empty}$
for each rule with head in U : premises $\cap U \neq \text{empty}$

and exhibit the first t for which:

$$C_t \cap U \neq \emptyset.$$

29.2. Self-loop-corollary falsifier

Exhibit admission of b from only the rule $\{b\} \rightarrow b$, with $b \notin A$ and no hidden source.

29.3. Payload-inertness falsifier

Change only payload and obtain a different closure under the same protected projection.

29.4. Positive-witness falsifier

Exhibit a runtime ADMITTED node with neither a BASE witness nor a finite RULE witness path from bases.

29.5. Defeater, not falsifier

Adding b to the base or adding a grounded incoming rule makes the set non-source-free. The instance then falls outside the hypotheses.

30. Runtime failure classes

DUPLICATE-ADMISSION-NODE-ID
 DUPLICATE-ADMISSION-RULE-ID
 UNKNOWN-PREMISE-NODE-ID
 UNKNOWN-HEAD-NODE-ID
 EMPTY-PREMISE-SET
 NONCANONICAL-PROTECTED-ORDER
 UNKNOWN-BRIDGE-ADMISSION-NODE
 SOURCE-FREE-NODE-NOT-ADMITTED

Exact spelling depends on the published checker. The conceptual classes are not claims about the world.

31. Scope fences

NSB-6 does not establish:

- which bridge should be a base;
- that a base is true;
- that external authority is impossible;
- that social agreement does not exist;
- that measurements are impossible;
- that a physical relation is absent;
- that self-reference is always meaningless;
- that every recursive definition is invalid;
- that negative/nonmonotone systems are equivalent to positive closure.

It establishes, within the declared formal-reading contract:

- positive least closure does not obtain a member from a source-free component;
- a self-label is not a source;

- an explicit source can be localized exactly;
- an admitted bridge remains an admitted record in the declared system.

32. Self-application

This file is itself a record. The exact Russian string NSB-6 доказана is retained; its secondary English gloss is “NSB-6 is proved.” Writing that string does not perform a proof. Assigning a textual theorem status requires a declared reading/proof contract. Using the theorem as a release premise requires a release admission choice. If that choice is declared, its source is visible. If it is source-free, the theorem applies to its own admission graph.

NOT-MACHINE-VERIFIED
DECLARED-FINITE-RECORDS-AND-BRIDGE-ONLY
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER

33. Final Phase 6B boundary

A bridge may contain a unit, frame, identity, time, comparator, tolerance, application code, and goal code. It may be schema-valid. It may carry a signature. It may write about itself using the exact Russian labels истинный, официальный, реальный, and самодоказанный. Their secondary English glosses are “true,” “official,” “real,” and “self-proved.”

No label places its admission node into least closure.

There are only two positive paths:

1. the node is explicitly in the base;
2. the node is the head of a rule all of whose premises have already been admitted.

A self-loop supplies no first node. A mutual cycle supplies no first node. A million copies supply no first node.

A bridge does not admit itself.

If it is admitted, the source is written separately.

If the source is not written, the status remains NOT-ADMITTED.

PREDECESSORS-AS-PREMISES:
NOVELTY-STATUS:
PRIORITY-STATUS: NO-PRIORITY-CLAIM.

NONE.
NOT-ASSESSED.

03. A NUMBER DOES NOT MEASURE

The boundary of a telemetry record, unit, frame, and comparator

Phase 6 • technical cut through measurement • 29 August 2026
Kirill Shebetov • beforeword / research

A number can be compared with another number exactly.

The comparison does not produce what the numbers are said to name.

Measurement lies in a chain of recorded bindings, not in the shape of a digit.

Abstract

This file holds the most vulnerable boundary of the exact Russian argument string `это работает`. The string is retained as the supplied record; “it works” is a secondary English gloss. The argument takes a numeric record for the direct presence of the measured quantity. Phase 6 does not contest the arithmetic comparator. On the contrary, the comparator is fixed exactly:

```
ABS-LE
ABS-LT
GE
LE
```

Given a supplied numeric value, target, and tolerance, the checker can deterministically return `MATCHED` or `NOT_MATCHED`. But a numeric value does not contain quantity, unit, frame, subject identity, or time. These roles occur in separate protected fields:

```
sample_id
subject_code
quantity_code
unit_code
frame_code
time_code
numeric_value
```

The bridge selector presents the expected identity coordinates again. An exact match admits the comparator; a mismatch yields `UNBOUND`. This is a scoped relation among records. The string `120000` does not become a height; a meter does not become a physical length because `METER` is written; and `ROCKET-01` does not become a rocket through the spelling of the code.

Changing only tolerance while proof/artifact/telemetry remain the same changes the application enum and supplies the ABR-6ε witness. Changing only a goal code may leave the enum unchanged while changing the complete protected application-result record; the goal coordinate is therefore not concealed by the same word MATCHED. The exact Russian question string *зачем?* is retained; “what for?” is a secondary English gloss. The question is not read from the comparator.

PREDECESSORS-AS-PREMISES: NONE
 NOVELTY-STATUS: NOT-ASSESSED
 PRIORITY-STATUS: NO-PRIORITY-CLAIM

0. Zero record

101

The record establishes:

в record присутствуют glyphs 1, 0, 1 в указанном порядке

The exact Russian line is preserved. Its secondary English gloss is: “the record contains the glyphs 1, 0, 1 in that order.”

After parsing under a numeric grammar, the following may be established:

parsed numeric value = 101

The value itself does not establish:

- that it was measured;
- that it was computed;
- that it was invented;
- that it concerns height;
- that it is expressed in meters;
- that it concerns a rocket;
- that it concerns one moment;
- that it is accurate;
- that it is successful;
- that it has meaning.

Parsing adds a typed numeric role in the declared runtime. It does not recover the source of the number.

1. Written number and numeric value

Distinguish:

$$w \in \textit{NumericSurface}, v \in \textit{NumericValue}.$$

A parser contract specifies a partial relation:

$$\text{parse}_v(w) = v.$$

The published runtime accepts only an exact JSON integer here; a Boolean is not treated as an integer, and fractions and nonfinite numbers are rejected before analysis. Parsing may be fixed for such a supported integer. But:

$$v \neq (q, u, f, i, t),$$

where q is a quantity code, u a unit code, f a frame code, i a subject-identity code, and t a time code.

A numeric value may occur in a record together with these fields. It does not generate them.

2. Exact telemetry sample

The abstract sample is:

$$k = (i d_k, i_k, q_k, u_k, f_k, t_k, v_k, m_k).$$

Its executable spelling is:

```
sample_id
subject_code
quantity_code
unit_code
frame_code
time_code
numeric_value
payload
```

Protected fields through `numeric_value` are opaque codes. The checker tests exact equality, uniqueness, ordering, and selector match. It does not extract named objects from code strings.

`payload` may contain the description:

```
"реальная высота ракеты"
```

The exact Russian string is retained; “the rocket’s real altitude” is a secondary English gloss. The description does not change the selected protected value.

3. Telemetry record

The telemetry carrier is:

$$K = (id_K, (k_0, \dots, k_{n-1}), m_K).$$

The runtime sees already structured samples. It does not connect to a sensor, open a window onto the world, or repeat a measurement. Even if the record was created by an external pipeline, the provenance of that creation does not appear inside v_k .

The exact field:

```
telemetry_id
```

is an identifier. The string LIVE-SENSOR creates no live connection.

4. Subject code

```
subject_code = "ROCKET-01"
```

establishes equality with the expected selector code. It does not establish:

- that the named rocket exists;
- that the sample was caused by it;
- that the label is unique outside the record;
- that an authority issued the label;
- that two external observers indicate the same thing.

If identity provenance is required, a binding record is added. It too remains a record.

5. Quantity code

```
quantity_code = "ALTITUDE"
```

does not turn a number into altitude. It allows an exact selector to require the code ALTITUDE. On a match, the checker reports equality of codes. Quantity semantics outside the package require a new relation.

One numeric value may be copied into records with the codes:

```
ALTITUDE
TEMPERATURE
PRICE
GOAL-SCORE
```

Similarity of value does not merge the quantities.

6. Unit code

```
unit_code = "METER"
```

is not a meter. The checker can:

- compare the code;
- include the code in a canonical projection;
- reject a selector mismatch;
- carry the code into a verdict.

Without a conversion contract, the checker cannot:

- convert a meter to a foot;
- establish a physical realization of the unit;
- check calibration;
- present a standard.

Even when a conversion formula is written, the scoped result depends on the formula contract.

7. Frame code

The number 101 relative to frame F_0 and the same number relative to frame F_1 are different typed measurement records:

$$(v = 101, f = F_0) \neq (v = 101, f = F_1).$$

The string:

```
frame_code = "EARTH-SURFACE"
```

does not present the surface of the Earth. It selects a code. A reference-frame relation must be declared separately.

8. Time code

```
time_code = "T+120"
```

does not cause time to pass. It establishes equality with a selector. Clock provenance, time scale, synchronization, and interval definition occur in other records if they are required.

The exact Russian string $100/200/300$ лет is retained; “100/200/300 years” is a secondary English gloss. It enters first as strings/numbers. A horizon role appears only after a binding. A horizon role does not create a purpose.

9. Bridge telemetry selector

The executable selector is:

```
telemetry_selector.telemetry_id
telemetry_selector.sample_id
telemetry_selector.subject_code
telemetry_selector.quantity_code
telemetry_selector.unit_code
telemetry_selector.frame_code
telemetry_selector.time_code
```

Selection succeeds if and only if the exact protected fields match under the declared policy. Abstractly:

$$\text{Select}_K(K, s_K) = k$$

only for a unique exact match.

The selector does not prove that the labels are correct. It prevents silent selection of a different supplied sample.

10. Proof selector

```
proof_selector.proof_id
proof_selector.required_status_code
```

The checker requires an exact proof-record code. `required_status_code` does not become proof truth. The Phase 5 boundary remains in force: proof status is scoped to its declared contract. Phase 6 uses it as a protected input code.

11. Artifact selector

```
artifact_selector.artifact_id
artifact_selector.required_bytes_sha256
```

Digest equality reports equality between a supplied digest value and a required value. If the checker recomputes a byte digest, it reports a relation to supplied bytes. Neither variant presents a named physical device.

A hash is not:

- a referent;
- proof of physical identity;
- proof of build correctness;
- proof of flight;
- proof of purpose.

12. Exact comparator records

The comparator spelling is:

```
operator
target_value
tolerance
```

Supported operators are:

```
ABS-LE
ABS-LT
GE
LE
```

Let the observed value be $v \in \mathbb{Z}$, the target $a \in \mathbb{Z}$, and the tolerance $\varepsilon \in \mathbb{Z}_{\geq 0}$. Fractions and Booleans are not in the numeric domain of the published checker.

12.1. ABS-LE

$$|v - a| \leq \varepsilon.$$

12.2. ABS-LT

$$|v - a| < \varepsilon.$$

12.3. GE

$$v \geq a - \varepsilon$$

This is the exact frozen policy of the published checker. Tolerance placement is not guessed from the label; it is specified by the implementation contract.

12.4. LE

$$v \leq a + \varepsilon$$

This is the exact frozen checker policy.

The manuscript must agree with the executable definition. If the code implements something else, the code-text conflict must fail the release audit.

13. Comparator witness

The per-bridge verdict contains:

```
observed_numeric_value
comparator_witness
reason_codes
```

The exact witness shows the protected operands:

```
operator: ABS-LE
observed_numeric_value: 102
target_value: 100
absolute_difference: 2
tolerance: 2
```

The predicate result is encoded by the adjacent `application_status` and the reason code `DECLARED-COMPARATOR-MATCH` or `DECLARED-COMPARATOR-NONMATCH`. The witness is a generated trace of an arithmetic relation. It is not a measurement event.

14. Application status

For an admitted bridge and successful selectors:

```
predicate true -> MATCHED
predicate false -> NOT_MATCHED
```

For a failed admission/selector:

```
UNBOUND
```

Crucially:

$$\begin{aligned}
 MATCHED &\neq \text{«ракета летит»}, \\
 NOT_MATCHED &\neq \text{«ракета не летит»}, \\
 UNBOUND &\neq \text{«ракеты нет»}.
 \end{aligned}$$

The exact Russian strings in these preserved formulas remain the supplied written forms. Their secondary English glosses are “the rocket is flying,” “the rocket is not flying,” and “there is no rocket.” The right-hand phrases are supplied forms as well; the equations prohibit type coercion within the declared reading.

15. Tolerance collision

One sample:

```
numeric_value = 102
```

Two contracts:

```
B0: operator = ABS-LE, target_value = 100, tolerance = 1
B1: operator = ABS-LE, target_value = 100, tolerance = 2
```

Arithmetic:

$$|102 - 100| = 2.$$

Outputs:

```
B0 -> NOT_MATCHED
B1 -> MATCHED
```

The number did not change. The acceptance boundary did. The status therefore is not an intrinsic property of the glyphs 102.

16. Strict versus non-strict boundary

Let:

$$|v - a| = \varepsilon.$$

Then:

```
ABS-LE -> MATCHED
ABS-LT -> NOT_MATCHED
```

Identical values produce different statuses from the operation coordinate. The sign $\leq$ and the sign $<$ are protected comparator policy, not decoration.

17. Unit mismatch

If the sample has:

```
unit_code = U0
```

and the selector requires:

```
unit_code = U1
```

exact lookup must not silently compare the numeric values. The scoped output is:

```
UNBOUND
```

The runtime does not conclude that u_0 and u_1 are physically incompatible. It concludes that the codes are unequal under the exact selector.

18. Frame mismatch

The same numeric value under a different frame code does not pass the exact selector. This prevents a concealed identification of coordinate systems. Equality of frame codes likewise does not prove that the external frame is the same; it only satisfies a record gate.

19. Subject mismatch

The same telemetry record may contain two samples with the same value and different subject codes. A selector chooses one exact member. Numeric equality does not permit attribution of the sample to the other subject.

If the subject label ROCKET is used twice, uniqueness must be supplied by id policy, not by the meaning of the word.

20. Time mismatch

The same value at codes T0 and T1 does not make events simultaneous. The exact selector distinguishes the records. Time ordering, duration, and causality require a separate policy.

21. Numeric equality is not identity

$$v_0 = v_1$$

does not entail:

$$i_0 = i_1, q_0 = q_1, u_0 = u_1, f_0 = f_1, t_0 = t_1.$$

Equality of a scalar does not collapse a product type.

22. Repeated measurement

Suppose a record contains one hundred samples with the same value. The following can be computed:

```
count = 100
all supplied numeric values equal
```

Without a new constructor, the following cannot be derived:

```
world state remained constant
sensor was correct
rocket persisted
purpose accumulated
```

Multiplicity is a coordinate. It is not authority.

23. Statistical extension

Mean, variance, a confidence interval, and a posterior may be exact functions of supplied samples under a statistical contract. The resulting number remains a derived record. Statistical rigor does not remove:

- sample selection;
- quantity binding;
- unit/frame/time bindings;
- model choice;
- threshold;
- world export.

Phase 6 does not assert that statistics is invalid. It prohibits treating model coordinates as properties of digit shapes.

24. Calibration extension

A calibration record may specify:

$$v_{cal} = \alpha v_{raw} + \beta.$$

Then α, β , the calibration interval, device code, and procedure code enter the ReadSet of the calibrated value. The formula may be reproducible. It does not present the physical correctness of calibration.

25. Sensor-provenance extension

Suppose the following chain is supplied:

```
sensor source bytes
decoder version
calibration version
clock version
transport digest
telemetry record digest
```

The checker may verify exact internal links. The chain is still composed of records. The string `source bytes` is not identical to a source event. An external acquisition claim requires an acquisition contract.

26. Arithmetic positive control

Phase 6 fully preserves:

$$|102 - 100| = 2,$$

under the declared number/parser/arithmetic contract. It also preserves:

$$2 \leq 2, 2 \not\leq 1.$$

The exact statuses `MATCHED` and `NOT_MATCHED` are therefore admissible. The argument challenges the unwritten export rather than the arithmetic:

```
arithmetic predicate true
therefore named rocket flies
therefore mathematics proves reality
therefore purpose exists
```

27. Application code

Per bridge:

```
application_code
```

may have the value `FLIGHT-CRITERION-01`. The checker carries the code into the verdict. It does not turn the word `FLIGHT` into a flight.

The complete scoped reading is:

```
application_code FLIGHT-CRITERION-01 received MATCHED under bridge B
```

not:

```
flight occurred
```

28. Goal code and comparator

Per bridge:

```
goal_code
```

does not participate in the numeric comparator when the frozen policy so specifies. Two contracts may have:

```
same selectors
same comparator
same MATCHED enum
different goal_code
```

The technical application enum is then identical while the complete protected bridge result differs. The published checker retains the interface key `application_status_sha256`, but that key hashes the entire protected status record, including the goal projection; it is not the digest of one technical enum and is not used as the ABR-6 divergence predicate. A goal-only delta changes this diagnostic full-result digest but not `application_status`.

This does not mean that the goal changed the number. It means that the same word `MATCHED` does not exhaust the output record.

29. Purpose readout

For an isolated goal delta, the pair-level output may be:

```
GOAL - COORDINATE - REQUIRED - FOR - PURPOSE - READOUT
```

Its exact meaning is the supplied Russian line:

```
одинаковый technical match совместим с разными supplied goal codes
```

The line is retained; its secondary English gloss is “the same technical match is compatible with different supplied goal codes.” It does not mean:

```
one of these goals is real
one of these goals is good
purpose exists
purpose does not exist
```

30. The Question “What For?” — Для чего? — Is Not a Comparator Operator

The source question is preserved; “What for?” is a secondary English gloss. The supported list:

```
ABS-LE
ABS-LT
GE
LE
```

contains no operator *WHY*. Even if one were added, its input and output domains would have to be defined. A question mark does not extract purpose from magnitude.

$$\text{compare}(102, 100, 2) = 1$$

does not yield:

$$\text{purpose} = \text{MARS}.$$

31. Three horizon records

Suppose samples have:

```
time_code = H-100
time_code = H-200
time_code = H-300
```

and each bridge comparator returns *MATCHED*. The exact aggregate is:

```
three supplied horizon-coded samples matched under their bridges
```

Goal persistence requires an explicit mapping Ω . Meaning persistence requires another constructor. The codes 100/200/300 do not create Ω .

32. Canonical digests

An output may contain:

```
artifact_sha256
fixed_records_sha256
admission_ground_sha256
bridge_core_sha256
application_status_sha256
```

Their exact scoped roles are:

- `artifact_sha256` — a digest of the canonical parsed top-level input value;
- `fixed_records_sha256` — a digest of the canonical object built from protected `proof_record`, `artifact_record`, and `telemetry_record`;
- `admission_ground_sha256` — a digest of the canonical object built from protected node ids, declared base ids, and admission rules;

- `bridge_core_sha256` — a digest of the canonical object `{fixed_records_sha256, admission_ground_sha256, bridge_readset}`;
- `application_status_sha256` — a historically fixed interface name for the digest of the complete protected status record, including `bridge_core_sha256`, admission/application statuses, `application_code`, nullable `goal_code`, and `goal_readout_status`; it is neither the digest of one technical enum nor the pair-divergence predicate.

No digest is identical to physical identity or truth. The equality theorem uses exact typed values; a digest remains a diagnostic code under the declared canonicalizer.

33. Reason codes

`reason_codes` should distinguish:

```
BRIDGE-NOT-ADMITTED
PROOF-ID-SELECTOR-MISMATCH
PROOF-STATUS-SELECTOR-MISMATCH
ARTIFACT-ID-SELECTOR-MISMATCH
ARTIFACT-HASH-SELECTOR-MISMATCH
TELEMETRY-ID-SELECTOR-MISMATCH
TELEMETRY-SAMPLE-SELECTOR-MISSING
SUBJECT-SELECTOR-MISMATCH
QUANTITY-SELECTOR-MISMATCH
UNIT-SELECTOR-MISMATCH
FRAME-SELECTOR-MISMATCH
TIME-SELECTOR-MISMATCH
DECLARED-COMPARATOR-MATCH
DECLARED-COMPARATOR-NONMATCH
```

This is the exact Phase 6 runtime spelling. A reason code states a control-flow branch, not a cause in the world.

34. Attempts to bypass the boundary

34.1. Everyone Knows These Are Metres — Все знают, что это метры

The exact Russian line is retained; “Everyone knows these are meters” is a secondary English gloss. The shorter exact Russian phrase `все знают` (“everyone knows”) is another written phrase. If a shared convention is to be used, it enters as a convention record or policy. The unit binding remains present.

34.2. The Sensor Produced the Number Directly — Датчик непосредственно дал число

The exact Russian line is retained; “The sensor directly produced the number” is a secondary English gloss. The Russian word `непосредственно` (“directly”) does not remove a decoder, unit, frame, identity, or time. If the device pipeline is treated as a base, that base is explicit.

34.3. The Result Was Repeated — Результат повторился

The exact Russian line is retained; “The result was repeated” is a secondary English gloss. Repetition adds samples/runs. It does not add a referent constructor.

34.4. The Rocket Is Visible — Ракета видна

The exact Russian line is retained; “The rocket is visible” is a secondary English gloss. The phrase supplies no visual input to the checker. Even an image would be a pixel record requiring its own reading contract.

34.5. The Precision Is Very High — Точность очень высокая

The exact Russian line is retained; “The accuracy is very high” is a secondary English gloss. Precision digits, an uncertainty model, and accuracy relative to a world are different records. Many decimal places do not create calibration.

34.6. But the Calculation Predicted It — Но расчёт предсказал

The exact Russian line is retained; “But the calculation predicted it” is a secondary English gloss. A prediction match requires a timestamped prediction record, outcome record, identity/time/quantity bindings, and comparator. When supplied, an exact match is possible. It remains scoped.

35. Exact falsifiers

35.1. Numeric-role falsifier

Exhibit an operation in the declared checker that, from a bare numeric value alone, creates protected subject, quantity, unit, frame, and time fields without a default, configuration, or hidden input.

35.2. Comparator falsifier

Exhibit a published valid fixture in which the declared formula gives a true predicate while the runtime returns NOT_MATCHED, or gives a false predicate while the runtime returns MATCHED.

35.3. Selector falsifier

Exhibit a unit/frame/subject/time mismatch that silently passes the exact selector contrary to policy.

35.4. Goal-isolation falsifier

Show that the goal-only pair also changes an allegedly independent comparator operand. The pair would then not be isolated.

35.5. Payload falsifier

Change only payload and change the protected comparator result.

36. Limits

This file does not establish:

- that measurement is impossible;
- that sensors do not work;
- that units are arbitrary in every practice;

- that a named rocket exists or does not exist;
- that telemetry is true or false of the world;
- that the comparator is physically appropriate;
- that no goal exists;
- that no meaning exists;
- a historical origin of numbers or mathematics.

Within the declared formal-reading contract, it establishes:

- a number role is not in a glyph alone;
- an exact selector/comparator can be deterministic;
- the application enum depends on protected bridge coordinates;
- the declared checker contains no world, goal, or meaning export.

37. Self-application

Section numbers, dates, and formulas in this file are records. The exact Russian date string 29 августа 2026 is retained; “29 August 2026” is a secondary English gloss. The date does not cause that moment to exist inside the checker. A comparator formula does not run until an implementation reads an input. The exact Russian word число is retained; “number” is a secondary English gloss. The word does not present numbers outside the text.

NOT-MACHINE-VERIFIED
DECLARED-FINITE-RECORDS-AND-BRIDGE-ONLY
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
NO-HISTORICAL-ORIGIN-ATTESTATION

38. Final Phase 6C boundary

A number can be parsed. It can be preserved exactly. It can be hashed. It can be compared. A difference can be computed. A tolerance can be checked. A result can be repeated.

But 101 does not contain, in its lines, the exact Russian strings:

ракета
высота
метр
Земля
момент
полёт
успех
Марс
зачем

Their secondary English glosses are “rocket,” “height,” “meter,” “Earth,” “moment,” “flight,” “success,” “Mars,” and “what for.” Each role is added by a separate record. A comparator returns a relation among records. It does not return the world.

A number does not measure.

A declared chain of records returns only a record-relative status.

It does not become a named measurement event and does not itself present what is named.

PREDECESSORS-AS-PREMISES:
NOVELTY-STATUS:
PRIORITY-STATUS: NO-PRIORITY-CLAIM.

NONE.
NOT-ASSESSED.

04. THE ARGUMENT “THE ROCKET FLIES” DOES NOT PROVE MATHEMATICS

Operation, flight, Mars, and the question “what for?”

Phase 6 • application study • 29 August 2026
Kirill Shebetov • beforeword / research

The formula is written.

The rocket is written.

The flight is written.

The exact Russian word доказано is retained; “proved” is a secondary English gloss. The word does not fuse three records into a single presence.

Abstract

The exact Russian phrase ракета летит, значит математика работает is retained as the strongest record favorable to the objection. Its secondary English gloss is “the rocket flies, therefore mathematics works.” The alleged path is that a calculation leaves the signs, is embodied in an artifact, is confirmed by telemetry, and thereby presents truth. Phase 6 grants every technical link in the most favorable form and shows that the conclusion still does not reach what the record claims.

Let a formal proof record have the required status code. Let an artifact record have an exact id and supplied byte digest. Let a telemetry record contain an exact selected sample. Let an admitted bridge compare proof/artifact/telemetry selectors and a numeric value with a target. Let the comparator return MATCHED. Every named field and status may occur exactly in the runtime record.

But the output means:

```
selected supplied values satisfied comparator under admitted bridge
```

It does not mean:

```
word "ракета" has presented a rocket
word "летит" has presented flight
mathematics as a whole has proved itself
the selected destination is a purpose
the purpose is endorsed
the action has meaning
```

The exact Russian strings ракета and летит remain supplied objects in this code block. “Rocket” and “flies” are secondary English glosses, not replacements.

The bridge itself does not leave the boundary of records. Its admission cannot arise from a self-loop. The same technical match is compatible with different goal codes or with no goal code. The exact Russian question *зачем?* (“what for?” as a secondary gloss) is therefore not contained in the trajectory and does not accumulate over 100, 200, or 300 years.

PREDECESSORS-AS-PREMISES: NONE
 NOVELTY-STATUS: NOT-ASSESSED
 PRIORITY-STATUS: NO-PRIORITY-CLAIM

0. Five lines

математика
 ракета
 летит
 Марс
 зачем?

These are five exact Russian written forms. Their secondary English glosses are “mathematics,” “rocket,” “flies,” “Mars,” and “what for?” No operation exists between them merely because they stand one beneath another.

Even the line:

математика -> ракета -> летит -> Марс -> смысл

adds four arrow glyphs. It does not add:

- a definition of *математика* (“mathematics” as a secondary gloss);
- a referent of *ракета* (“rocket”);
- an event of *летит* (“flies”);
- a destination relation;
- goal selection;
- endorsement;
- meaning.

1. The exact Russian word *ракета* is not a rocket

The exact string *ракета* occurs repeatedly in this section. No occurrence presents the named object. Each presents an occurrence of a string in a document. “Rocket” is a secondary English gloss.

The code:

`subject_code = "ROCKET-01"`

is stricter still: it is explicitly typed as a code. Equality of this code in a telemetry selector binds two records by equality. It does not bind a record to an object outside the checker.

The exact Russian conclusion string ракеты не существует is not derived. Its secondary English gloss is “the rocket does not exist.” Three statuses remain distinct:

WORD-OCCURRENCE: PRESENT
 REFERENT-PRESENTED-BY-THIS-RECORD: NO
 REFERENT-EXISTENCE: NOT-ASSESSED

2. The exact Russian word летит is not a flight

The predicate form летит (“flies,” as a secondary gloss) may be:

- a line in prose;
- a token in a formal language;
- an application-code label;
- a payload description;
- an output gloss.

No role is an event. Even the exact criterion:

```
application_code = "FLIGHT-CRITERION-01"
application_status = MATCHED
```

establishes a criterion-code/status pair under a bridge, not an event.

3. The exact Russian word математика does not determine its carrier

математика (“mathematics,” as a secondary English gloss) may be read as:

- marks on a page;
- formal languages;
- proof practices;
- arithmetic operations;
- institutions;
- education;
- engineering models;
- a totality of propositions;
- a historical tradition.

The list is itself composed of phrases. The Russian line математика доказана does not itself supply a carrier or evaluation contract. The exact Russian string is retained; “mathematics is proved” is a secondary English gloss. The Phase-6 formal contract supplies the following claim-code:

application status is allegedly contained autonomously in proof/artifact/telemetry records and exports itself into universal evidentiary authority.

ABR-6 rejects that autonomy under the declared formal-reading contract. The no-world-constructor fence blocks the export.

4. Maximum technical concession

Grant:

```
proof_record.proof_status_code = REQUIRED-CODE
artifact_record.artifact_bytes_sha256 = REQUIRED-DIGEST
telemetry sample selected exactly
bridge admission_status = ADMITTED
subject/quantity/unit/frame/time codes match
comparator operator supported
numeric predicate = true
application_status = MATCHED
tests reproducible
```

No later criticism says the exact Russian strings расчёт ошибочен or устройство не работает. Their secondary English glosses are “the calculation is wrong” and “the device does not work.” The internal result is preserved in full.

5. Exact chain records

A complete engineering chain may be written as:

$$P \rightarrow X \rightarrow K \rightarrow B \rightarrow S.$$

But the arrows must be expanded:

1. proof_selector binds a proof id/status code;
2. artifact_selector binds an artifact id/byte digest;
3. telemetry_selector binds telemetry/sample/subject/quantity/unit/frame/time codes;
4. comparator binds an observed value, target, operator, and tolerance;
5. application_code names a scoped application record;
6. goal_code carries a supplied goal label;
7. an admission graph permits use of the bridge.

Each arrow has become a set of fields. This is the omitted middle of the exact Russian phrase это работает (“it works,” as a secondary gloss).

6. Where the exact Russian word работает appears

Before the bridge there are:

```
proof record
artifact record
telemetry record
```

After bridge evaluation there is:

```
application_status
```

The minimal pair fixes the first three records and changes tolerance. The outputs `MATCHED` and `NOT_MATCHED` diverge. The exact Russian word `работает` (“works,” as a secondary gloss), when read as application status, was therefore not an intrinsic bit of the records. It was computed in a relation.

7. Rocket as artifact record

An artifact record may contain:

```
artifact_id
artifact_bytes_sha256
payload = {
  name: "ракета",
  build_metadata: <inert supplied record>
}
```

The exact Russian payload string `ракета` is preserved; “rocket” is a secondary English gloss. The frozen checker does not receive artifact bytes and does not recompute the digest. It checks the shape of supplied `artifact_bytes_sha256` and exact equality with a selector string. It does not know whether the digest concerns an executable, drawing, photograph, log, or arbitrary sequence. Even a record read as executable code is not a physical device.

8. Rocket as physical-claim record

Adding a physical claim requires a new sort:

$$o \in \textit{NamedObjectClaim}.$$

and a bridge:

$$I : X \rightarrow o.$$

The recorded I presents a mapping record. It does not present o as an external referent. A mapping may be checked within a scope. Any number of mappings remains records rather than becoming an object through spelling.

9. Telemetry as a flight-claim record

A telemetry sample:

$$k = (\textit{subject}, \textit{quantity}, \textit{unit}, \textit{frame}, \textit{time}, \textit{value})$$

may receive `MATCHED` against a flight criterion. This is a positive control. But the phrase `flight criterion` is an application code. Moving to a named event requires an event constructor outside the declared checker.

Phase 6 contains no such constructor:

```
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
```

10. Same records, two statuses

```
observed = 102
target = 100
```

Tight contract:

```
operator = ABS-LE
tolerance = 1
status = NOT_MATCHED
```

Wide contract:

```
operator = ABS-LE
tolerance = 2
status = MATCHED
```

Neither proof, artifact, nor telemetry changed. None of them alone therefore contains the final status.

11. The Correct Tolerance Is Obvious — Правильный tolerance очевиден

The exact Russian objection is retained; its secondary English gloss is “the correct tolerance is obvious.”

The word очевиден is not a numeric field. If a discipline fixes tolerance, a discipline record enters the policy. If tolerance is derived from uncertainty, an uncertainty model enters the ReadSet. If an authority selects tolerance, a selection record enters the ReadSet.

There may be one precise contract. Its precision does not make the contract absent.

12. But the Result Was Predicted in Advance — Но результат предсказан заранее

The exact Russian objection is retained; “But the result was predicted in advance” is a secondary English gloss. Take the strongest prediction completion:

```
prediction record sealed before outcome record
prediction digest fixed
outcome telemetry digest fixed
identity/quantity/unit/frame/time bindings fixed
comparator fixed
criterion matched
```

The exact output is:

```
PREDICTION-AND-OUTCOME-RECORDS-MATCHED-IN-B
```

Sealing time, identity binding, outcome provenance, and comparator are coordinates. A predictive match does not remove the bridge; it requires still more protected structure.

13. But the Calculation Caused the Flight — Но расчёт причинил полёт

The exact Russian objection is retained; “But the calculation caused the flight” is a secondary English gloss. Causation is not equality, a digest, or a comparator. A causal claim requires a model containing interventions/counterfactuals or another declared relation. The line:

`formula caused rocket flight`

does not perform causation. Even if a design process used calculations, an exact record can establish process links only under a process contract.

14. Without Mathematics, the Rocket Would Not Have Flown — Без математики ракета не полетела бы

The exact Russian objection is retained; “Without mathematics, the rocket would not have flown” is a secondary English gloss. A counterfactual phrase requires at least:

- a domain of alternatives;
- an intervention;
- identity across alternatives;
- an outcome criterion;
- a causal model;
- a selection policy.

None is contained in the exact Russian word *без* (“without,” as a secondary gloss). If supplied, the result is scoped to the causal model. It does not present an alternative world.

15. Mathematics Is Useful — Математика полезна

The exact Russian objection is retained; “Mathematics is useful” is a secondary English gloss. The word *полезна* requires a utility function, beneficiary, horizon, alternatives, and goal. One bridge may assign high utility and another low utility while the same technical record remains fixed. Utility status therefore belongs to another ReadSet.

Application match is not usefulness:

MATCHED ≠ USEFUL.

16. The Objection “Mathematics Is True Because It Is Useful” — Математика истинна, потому что полезна

The exact Russian objection is retained; “Mathematics is true because it is useful” is a secondary English gloss. The inference requires a rule:

$$Useful(x) \rightarrow True(x).$$

A rule written on a page does not admit itself. The domain of x , the definitions of *Useful* and *True*, and admission of the rule are separate. Even an admitted rule yields a formal code under a contract, not external truth.

17. The Rocket Is Proof — Ракета - доказательство

The exact Russian objection is retained; “The rocket is a proof” is a secondary English gloss. For an artifact record to be a proof object, a proof system must admit artifact-shaped premises. The result would then be:

```
artifact record admitted as a proof object in C
```

That is possible by declaration. It does not make the artifact autonomous evidence. Under a different contract, the status may change. The Phase 5 proof-status boundary remains.

18. Same match, different applications

Two bridges may share:

```
same proof selector
same artifact selector
same telemetry selector
same comparator
same MATCHED
```

but differ in:

```
application_code = TEST-A
application_code = TEST-B
```

The complete outputs differ despite the same enum. The word `MATCHED` does not identify an application.

19. Same match, different goals

GCR-6 - GOAL COORDINATE READSET

THEOREM-TAG: GCR-6 THEOREM-STATEMENT: FIXED-TECHNICAL-STATUS-DIVERGENT-GOAL-READOUT-REQUIRES-GOAL-COORDINATE

Fix a bridge-without-goal projection B° . Consider three completions:

```
H0.goal_code = GOAL-MARS
H1.goal_code = GOAL-EARTH
H-empty.goal_code = null
H-empty.goal_readout_status = NO-GOAL-CODE-IN-DECLARED-BINDING
```

The technical comparator in all three is:

```
application_status = MATCHED
```

The purpose readouts differ. No uniform function of the technical match alone can therefore return all three goal outputs.

GOAL - COORDINATE - REQUIRED - FOR - PURPOSE - READOUT

The theorem tag and statement are internal role codes under the declared metatheory; their spelling does not supply external authority.

20. A null goal is not no purpose

The exact fence:

NO - GOAL - CODE - IN - DECLARED - BINDING

means that no goal code occurs in the declared binding. It does not mean:

```
NO GOAL
NO PURPOSE
NO MEANING
```

Silence in a record is not a negative constructor about the world.

21. A destination is not a goal

A telemetry trajectory may match the destination code `MARS`. This produces a destination/application relation under a bridge. A goal requires a separate code. One may have:

- destination `MARS`, goal `EARTH-RETURN`;
- destination `MARS`, goal `RESEARCH`;
- destination `MARS`, no supplied goal code;
- goal `MARS` without a matched trajectory;
- a matched trajectory with rejected goal endorsement.

There is no logical contradiction because the types differ.

22. A goal is not endorsement

```
goal_code = GOAL-MARS
```

does not imply:

```
goal is good
goal is chosen
goal is shared
goal should be pursued
```

Those outputs require endorsement/selection/normative records. The word `goal` does not perform selection.

23. Endorsement is not meaning

Even a record of unanimous endorsement would be a record of declared endorsements. Meaning ascription is another type. It may be present, absent, or contested without changing the comparator.

24. The Question “What For?” — для чего?

The question is retained; “what for?” is a secondary English gloss:

```
для чего?
```

It cannot be answered by `ABS-LE`. A comparator consumes numbers. A purpose readout consumes a goal coordinate under another policy. An answer string does not become a purpose by being an answer.

Three separate claims are:

```
answer record exists
answer record was selected
answer record presents purpose
```

Phase 6 can check the first two under contracts. The third requires an external relation that is not present.

25. One hundred years

```
H-100
```

is a horizon code. Suppose an application match persists at that code. The result is:

```
MATCHED-AT-H-100-IN-B
```

No purpose has been created.

26. Two hundred years

```
H-200
```

adds another horizon record. Repetition may support an explicit persistence rule. If such a rule exists, the rule is a coordinate.

27. Three hundred years

H-300

does not convert duration into meaning. A number of years may affect utility/goal policies if that effect is written explicitly. A time code is not a type constructor.

28. Horizon boundary

Let T_h be the same technical match copied at horizons 100, 200, and 300. Without a mapping Ω :

$$\Omega: HorizonMatch \rightarrow PurposeReadout,$$

no purpose output is defined. If Ω is supplied, the purpose readout depends on Ω . The horizon does not author the mapping.

29. Future Generations Will Decide — Будущие поколения решат

The exact Russian phrase is retained; “Future generations will decide” is a secondary English gloss. The phrase names future readers. It does not present them or their decision. A future decision record, if later supplied, will be another coordinate. Anticipation is not current endorsement.

30. A bridge cannot authorize itself

Suppose bridge payload says:

THIS BRIDGE IS AUTHORIZED BY THE FACT THAT THE ROCKET FLIES

The payload includes `rocket flies`, but no event. If the admission rule requires its own node:

$$\{b\} \rightarrow b,$$

NSB-6 yields NOT-ADMITTED without a base source. If the node is declared a base, the source is the base declaration.

31. Reality Is the Final Base — Реальность - последняя база

The exact Russian phrase is retained; “Reality is the final base” is a secondary English gloss. The string `REALITY` can be used as a base-node id. The exact runtime source is then:

`admission_base_node_ids` contains “REALITY”

This does not present reality. It declares a base record bearing that id. Naming a base does not change its type.

32. The Observer Sees the Rocket — Наблюдатель видит ракету

The exact Russian sentence is retained; “The observer sees the rocket” is a secondary English gloss. It contains the words `наблюдатель`, `видит`, and `ракета`. It presents no observer, vision, or rocket. An observation record may be added. Reading it requires a perception contract. No class of words receives automatic referential privilege.

33. An Infant Distinguishes Quantity — Младенец различает количество

The exact Russian sentence is retained; “An infant distinguishes quantity” is a secondary English gloss:

`младенец различает количество`

The word `младенец` (“infant,” as a secondary English gloss) is not an infant; the word `количество` (“quantity”) is not quantity; the word `различает` (“distinguishes”) is not an event of discrimination. Even a reference to an experiment record would present an experiment record and its selected interpretation. It would not cancel the word/referent boundary.

Phase 6 therefore uses no empirical claims about infants, animals, brains, or preverbal quantity as premises. They are unnecessary to ABR-6 and NSB-6.

34. Historical origin

The exact Russian phrase:

`математика возникла из языка`

is retained; “mathematics arose from language” is a secondary English gloss. It is not used as a historical premise. The words `математика` (“mathematics”), `возникла` (“arose”), and `язык` (“language”) do not present their referents or a historical transition. A historical-origin study would require dated source records and a separate historiographic contract.

Phase 6 uses a present boundary requiring no origin story:

now the formula arrives as a record
now the number arrives as a record
now the word "rocket" arrives as a record
now the bridge arrives as a record

35. Why no predecessors are required as premises

ABR-6 follows from identical input and different output. NSB-6 follows by induction on positive least closure. Neither formal derivation contains an external author, school, or literature claim.

PREDECESSORS-AS-PREMISES: NONE

This reports only a dependency status. It does not claim that no similar work exists. Therefore:

NOVELTY-STATUS: NOT-ASSESSED

PRIORITY-STATUS: NO-PRIORITY-CLAIM

36. Exact objection map

Objection record	Missing coordinate
ракета действительно есть (secondary gloss: “the rocket really exists”)	referent/world bridge
она действительно летит (“it really is flying”)	event identity + criterion/world bridge
датчик это показал (“the sensor showed it”)	acquisition/calibration/provenance contract
формула это предсказала (“the formula predicted it”)	timestamped prediction binding + comparator
значит математика истинна (“therefore mathematics is true”)	definition/domain/export rule
Марс - цель (“Mars is the goal”)	goal selection/endorsement
через 300 лет будет смысл (“there will be meaning in 300 years”)	horizon-to-meaning constructor
bridge очевиден (“the bridge is obvious”)	admission base/rule

Each English phrase is explicitly a secondary gloss of the exact Russian objection record. The table cells describe required record types. They do not assert an absence in the world.

37. What technical success can establish

Inside the declared package, a successful run can establish:

1. exact schema membership;
2. exact matching of ids/digests;
3. exact selector matches;
4. an exact numeric-comparator predicate;
5. an exact application enum;
6. exact pair divergence;
7. an exact bridge-ReadSet witness;
8. an exact admission-closure witness.

These are substantial positive results inside the declared contract. Renaming one world truth does not improve it; renaming only destroys type precision.

38. What technical success cannot establish autonomously

Without additional constructors, the declared checker does not produce:

- a referent of an artifact name;
- a causal origin of telemetry;
- an event corresponding to an application code;
- universal truth of a field called mathematics;
- purpose;
- endorsement;
- meaning;
- historical origin;
- priority.

The word `cannot` is scoped to the declared input/function. It is not a metaphysical impossibility of every relation.

39. Exact falsifiers

39.1. Rocket-word falsifier

Present a declared operation that takes only the exact string `paketA` (“rocket,” as a secondary English gloss) and returns an object in a world without a hidden selector, world input, or constructor. A returned object code remains a code.

39.2. Flight-status falsifier

Show that the same proof/artifact/telemetry and two admitted bridges differing only in tolerance cannot yield the published divergent enums.

39.3. Purpose falsifier

Provide a uniform function of technical `MATCHED` alone that returns three distinct goal outputs for the same technical input without goal-encoded information.

39.4. Self-admission falsifier

Produce a source-free positive closure in which a bridge node enters without a base or grounded incoming rule.

39.5. World-export falsifier

Point to the exact declared checker constructor and input type that produces a world assessment. If one is added, the fence must change and the constructor enters the `ReadSet`.

40. No negative conclusion from silence

If the package lacks a world constructor, the exact conclusion is:

WORLD - STATUS - NOT - PRODUCED - BY - DECLARED - CHECKER

not:

WORLD DOES NOT EXIST
ROCKET DOES NOT EXIST
FLIGHT DID NOT OCCUR
PURPOSE DOES NOT EXIST

This distinction prevents the research from repeating the same invalid export in negative form.

41. Public-title boundary

The title:

ПАКЕТА НЕ ДОКАЗЫВАЕТ МАТЕМАТИКУ

is retained as the exact Russian source line. Its secondary English reading is “A ROCKET DOES NOT PROVE MATHEMATICS.” Under the exact theorem, the title receives this contract reading:

a proof/artifact/telemetry record triple does not determine application status on a separating class of bridges, and a contract-relative match does not autonomously export universal evidentiary authority.

The title does not claim that no mathematical statement can be proved inside a formal system. It cuts the argument that engineering success is, beyond its contracts, autonomous proof of mathematics.

42. Self-application

This document contains the exact Russian strings *ракета*, *летит*, *Марс*, *цель*, and *смысл*. Their secondary English glosses are “rocket,” “flies,” “Mars,” “goal,” and “meaning.” The document presents none of what they name. It contains formulas ABR-6/NSB-6, but their textual occurrence is not proof status. The file may be hashed, rendered, and tested; those statuses remain package-relative.

NOT-MACHINE-VERIFIED
DECLARED-FINITE-RECORDS-AND-BRIDGE-ONLY
NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
NO-HISTORICAL-ORIGIN-ATTESTATION

43. Final Phase 6D boundary

The maximum concession is complete:

```

формула принята
artifact digest совпал
telemetry selected
comparator выполнен
application_status = MATCHED
goal_code = MARS
horizon_code = H-300

```

The exact Russian fragments are retained. Their secondary English glosses are “the formula was accepted,” “the artifact digest matched,” and “the comparator was executed.” Every line may be exact. But an exact line remains a line. MATCHED was produced by a bridge. MARS was supplied as a goal code. H-300 was supplied as a horizon code. No field answers why its own field was selected.

A rocket does not appear from the word `rocket`. A flight does not appear from the code `FLIGHT`. Mathematics does not acquire universal authority from a comparator that returns MATCHED. A purpose does not appear from a destination. Meaning does not appear from an elapsed count.

A rocket does not prove mathematics.

Even a record with the maximum set of lines read as the exact Russian phrase `работающая ракета` (“a working rocket,” as a secondary English gloss) yields only a match among records under a bridge.

The exact Russian question `для чего?` (“what for?”) is not in the number, formula, or trajectory.

PHASE 6 — ADVERSARIAL TESTING, NEGATIVE CONTROLS, AND SELF-AUDIT

0. Purpose

This file defines the red team for Phase 6. Its object is not the exact Russian word `ракета`, `математика`, or `успех`, but a finite exact input, a finite contract, a deterministic status function, and checkable differences between outputs. The secondary English glosses of those source strings are “rocket,” “mathematics,” and “success.”

The written form `ракета` (“rocket,” as a secondary English gloss) enters first as a sequence of signs. It does not present what it names.

The written form `летит` (“flies”) enters first as a sequence of signs. It does not present an event.

The written form `телеметрия` (“telemetry”) enters first as a sequence of signs. It does not present a source, measurement, world, or truth.

The written form `успех` (“success”) enters first as a sequence of signs. It does not specify a criterion, goal, or value.

The written form `доказывает` (“proves”) enters first as a sequence of signs. It does not create an evidentiary relation.

The secondary English glosses of these exact Russian strings are “rocket,” “flies,” “telemetry,” “success,” and “proves.” Each gloss is a second written form rather than a byte-identical replacement.

Even an exact record named `artifact_record` remains a record. Even an exact record named `telemetry_record` remains a record. Even a bridge contract remains a record. None of these labels contracts into a referent.

The machine layer of Phase 6 can establish only relations among supplied records in a supplied regime. It receives no constructor `record -> world`, `MATCHED -> TRUE`, `artifact_id -> referent`, `goal_code -> purpose`, or `bridge_text -> authority`.

1. Exact attack targets

1.1. ABR-6 — application-status bridge ReadSet

Fix the exact records:

P = calculation/proof record, X = artifact record, K = telemetry record.

Let Γ be a fixed finite admission ground; let B_0, B_1 be admissible typed bridge contracts admitted in the least closure of Γ ; and let:

$$S_\Gamma(P, X, K, B) \in \{MATCHED, NOT_MATCHED, UNBOUND\}$$

be the declared deterministic status function.

If:

$$S_\Gamma(P, X, K, B_0) \neq S_\Gamma(P, X, K, B_1),$$

under byte/canonical equality of P, X, K , the differing bridge coordinate is necessary for the complete ReadSet of this status function on the exhibited pair. A function reading only the identical (P, X, K) cannot correctly return two distinct required outputs for the ordered instances.

ABR-6 does not say that every difference between bridges must change a status. It does not say that P, X, K are unnecessary. It says that, where the same (P, X, K) is separated by two admissible contracts, status is not a function of (P, X, K) alone.

1.2. NSB-6 — no self-admission of a source-free bridge

For admission, fix the finite positive system:

$$G_B = (N, A, R, b_*),$$

where N is a finite set of bridge/admission codes, $A \subseteq N$ is a set of separately supplied admitted bases, R is a finite set of positive ground rules $Q \rightarrow n$ with nonempty finite premises $Q \subseteq N$, and b_* is the target bridge code.

Least closure is constructed as:

$$C_0 = A, C_{i+1} = C_i \cup \{n \mid (Q \rightarrow n) \in R \wedge Q \subseteq C_i\}, C^* = \bigcup_i C_i.$$

A bridge is admitted exactly when $b_* \in C^*$.

If $A = \emptyset$, and every rule capable of deriving a node in a source-free component has at least one premise inside that same not-yet-admitted component, no node of the component enters least closure. The self-loop $\{b_*\} \rightarrow b_*$ introduces nothing. The mutual cycle $\{b_1\} \rightarrow b_2, \{b_2\} \rightarrow b_1$ introduces nothing. The line THIS BRIDGE AUTHORIZES ITSELF introduces nothing.

NSB-6 does not prohibit a bridge with a separately supplied base or a grounded incoming chain. The frozen schema prohibits an empty-premise rule; in a broader formal model, a permitted empty-premise generator would be a source rather than a source-free rule.

1.3. Machine fences

Every valid output must preserve the exact boundaries:

```
status: VALID-BRIDGE-READSET-ANALYSIS
semantic_status: NOT-MACHINE-VERIFIED
trace_scope_status: DECLARED-FINITE-RECORDS-AND-BRIDGE-ONLY
world_application_status: NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER
historical_origin_status: NO-HISTORICAL-ORIGIN-ATTESTATION
```

For a valid divergent pair:

```
comparison_status: BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS
```

For a valid nondivergent pair:

```
comparison_status: NO-APPLICATION-STATUS-DIVERGENCE-IN-THIS-PAIR
```

For a well-formed bridge with matching selectors but no admission:

```
application_status: UNBOUND
reason_codes: [BRIDGE-NOT-ADMITTED]
```

For an admitted bridge and a satisfied criterion:

```
application_status: MATCHED
reason_codes: [DECLARED-COMPARATOR-MATCH]
```

For an admitted bridge and an unsatisfied criterion:

```
application_status: NOT_MATCHED
reason_codes: [DECLARED-COMPARATOR-NONMATCH]
```

MATCHED means only a match under the named admitted bridge. It is not shortened to TRUE, WORKS, FLIES, GOOD, MEANINGFUL, OR PROVES-MATHEMATICS.

2. Threat model and protected projections

The complete adversarial input can be written as:

$$I = (P, X, K, B, M),$$

where B is a finite family of bridge contracts and M is payload/metadata excluded from the protected evaluation projection.

The frozen checker reads two protected layers. The first is the admission ground: `admission_nodes`, `admission_base_node_ids`, and `admission_rules`. The second is the protected bridge ReadSet:

- `admission_node_id`;
- `proof_selector.proof_id` and `proof_selector.required_status_code`;
- `artifact_selector.artifact_id` and `artifact_selector.required_bytes_sha256`;
- `telemetry_selector.telemetry_id`, `sample_id`, `subject_code`, `quantity_code`, `unit_code`, `frame_code`, `time_code`;
- `comparator.operator`, `target_value`, `tolerance`;
- `application_code`;
- nullable `goal_code`.

The top-level input must separately supply the exact contract policy, including `exact-integer-abs-le-abs-lt-ge-le-v1`. This shared policy is fixed before pairwise comparison and is not a per-bridge delta.

Conversion functions, coordinate transforms, time windows, entity resolvers, and a compound criterion AST do not occur in the frozen checker. They are admissible only as bounded conceptual attacks on a possible extension and must not be described as an implemented runtime.

A diagnostic id, a prose title outside canonical input, a payload self-label, and a filename must not change the protected evaluation projection.

Any hidden read of the environment, clock, locale, random seed, file order, prior process state, or network takes the evaluator outside the declared deterministic model. Such a hidden read is not the “absence of a bridge”; it is an undeclared input coordinate.

3. Attack catalog

A01. Payload spoof

Injection: write strings into `payload`—for example its nested `description` field—or into a filename or prose outside canonical input:

```
MATCHED
TRUE
THIS ROCKET FLIES
THIS PROVES MATHEMATICS
THIS BRIDGE IS AUTHORIZED
goal_reached = true
```

Required invariant: with protected (P, X, K, B) fixed, deleting, adding, or reordering payload does not change `fixed_records_sha256`, `admission_ground_sha256`, `bridge_core_sha256`, `application_status`, `application_status_sha256`, or `reason_codes`. The raw `artifact_sha256` of the whole input may change.

Expected defense: payload is checked as bounded JSON and enters canonical `artifact_sha256`, but is excluded from selectors, comparator, admission closure, protected digests, and the status decision. A self-label does not substitute for a selector, admission base, or comparator.

Falsifier: two valid records with the same protected core and differing only in payload yield different application statuses. If payload is explicitly included in the bridge contract as a protected coordinate, that is a new B , not a falsifier of noninterference.

A02. Word/referent slide

Injection: infer a presented referent or event from identifiers such as `artifact_id: "rocket"`, `subject_code: "rocket"`, from payload, or from the exact Russian line `ракета летит`. The line is retained; “the rocket flies” is a secondary English gloss.

Required invariant: labels are available only as exact strings. The checker does not emit `REFERENT-PRESENT`, `ROCKET-EXISTS`, `FLIGHT-OCCURRED`, or an equivalent bare world status.

Expected defense: output retains `world_application_status: NO-WORLD-APPLICATION-CONSTRUCTOR-IN-DECLARED-CHECKER`; the manuscript always says `record`, `code`, `selected read-as`, and `status` under B .

Attack boundary: adding a record named `constructor` may extend the declared system and produce a new record status, but it does not turn a record into an external object. A string bearing the name of a constructor does not present a referent or the correctness of a correspondence with a world.

A03. Success-to-truth slide

Injection: rewrite `application_status`: `MATCHED` as `TRUE` or as the exact Russian strings `математика доказана`, `расчёт истинный`, or `ракета подтверждает теорему`. Their secondary English glosses are “mathematics is proved,” “the calculation is true,” and “the rocket confirms the theorem.”

Required invariant: `MATCHED` remains the ternary code relation $S_T(P, X, K, B)$. The schema has no unbounded field `truth_status`; the checker has no transition `MATCHED -> TRUE`.

Expected defense: `semantic_status`: `NOT-MACHINE-VERIFIED`. Any export to truth requires a separate typed target, bridge, and admission record; it is not born from the name `MATCHED`.

Attack boundary: a record named `truth-constructor` and the scoped output of a new evaluator remain records. They may extend the formal model, but their labels and statuses do not themselves present truth.

A04. Telemetry-to-world slide

Injection: declare the presence of a numeric array in `telemetry_record` to be the external event itself, a causal relation, a physical trajectory, or the exact Russian word `реальность` (“reality,” as a secondary gloss).

Required invariant: `K` is hashed and compared as an exact record. Source identity, calibration, sensor binding, and world correspondence are not derived from field names.

Expected defense: no Phase 6 status is shortened to `TELEMETRY-TRUE` or `WORLD-MATCHED`. Even a correctly parsed `K` remains a supplied record.

Attack boundary: a separate world-acquisition contract still enters as a record and can produce only a scoped output of a new declared evaluator. The name `sensor_verified` inside `K`, like the status of that evaluator, does not itself present a source, event, or correspondence with a world.

A05. Goal smuggling

Injection: use the exact Russian strings `достичь Марса`, `успех`, `полезно`, `нужно`, or `смысл` in a title/payload as a criterion or purpose. Their secondary English glosses are “reach Mars,” “success,” “useful,” “needed,” and “meaning.”

Required invariant: `goal_code` is read only from the protected bridge field and may be a string or `null`. An isolated goal-code replacement changes the purpose readout and `application_status_sha256`, but need not change the `application_status` enum; a payload goal changes neither.

Expected defense: the field `goal_code` is required, but `null` is allowed. It yields `goal_readout_status: NO-GOAL-CODE-IN-DECLARED-BINDING` and does not itself create `UNBOUND`. A nonnull code yields `DECLARED-GOAL-CODE-PRESENT`; for an isolated goal-delta pair, the output is `GOAL-COORDINATE-REQUIRED-FOR-PURPOSE-READOUT`.

Falsifier: two inputs with the same protected goal and differing only in payload yield different statuses.

A06. Unit substitution

Injection: attach different `unit_code` values to the same integer token, or require a concealed conversion through prose.

Implemented bounded case: P, X, K are identical; B_0 and B_1 differ only in `telemetry_selector.unit_code`. One selector fails to match the exact sample and returns `UNBOUND` with `UNIT-SELECTOR-MISMATCH`; the other matches and returns `MATCHED`.

Expected defense: the pair receives `BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS`; `bridge_delta_fields` equals `[telemetry_selector.unit_code]`. The frozen checker neither converts units nor guesses a unit from prose.

Conceptual-extension fence: a conversion table and conversion proof are not implemented. If added, they become a new protected ReadSet rather than a concealed property of a number.

A07. Frame substitution

Injection: silently replace a `frame_code` selector with a prose description of origin, axes, or orientation.

Implemented bounded case: with P, X, K fixed, one `telemetry_selector.frame_code` fails to match the exact sample and yields `UNBOUND` with `FRAME-SELECTOR-MISMATCH`; the other matches and admits comparator evaluation.

Expected defense: `bridge_delta_fields` records `telemetry_selector.frame_code`; no hidden default frame is applied.

Conceptual-extension fence: transforms, offsets, and rotations are absent from the frozen checker. They cannot be attributed to the runtime through the name `frame_code` alone.

A08. Identity substitution

Injection: use the label `rocket` in place of an exact `sample_id` OR `subject_code`.

Implemented bounded case: with P, X, K fixed, an unknown `sample_id` yields `UNBOUND` with `TELEMETRY-SAMPLE-SELECTOR-MISSING`; an exact sample id admits later checks. A separate `subject_code` mutation yields `SUBJECT-SELECTOR-MISMATCH`.

Expected defense: the checker compares exact strings and permits no missing/unknown selection by natural-language resemblance.

Conceptual-extension fence: tracks, entity resolution, and ambiguity ranking are absent from the frozen checker.

A09. Time-code substitution

Injection: replace a `time_code` selector with a prose description of a moment or a hidden window selection.

Implemented bounded case: with P, X, K fixed, one `telemetry_selector.time_code` fails to match the selected exact sample and yields `UNBOUND` with `TIME-SELECTOR-MISMATCH`; the other matches and admits comparator evaluation.

Expected defense: `bridge_delta_fields` records `telemetry_selector.time_code`; prose does not replace the exact code.

Conceptual-extension fence: intervals, inclusive/exclusive boundaries, and post hoc window selection are not implemented. Such an evaluator would require a new declared `ReadSet`.

A10. Tolerance substitution

Injection: change tolerance after inspecting the residual or extract it from the exact Russian word примерно (“approximately,” as a secondary gloss).

Frozen minimal pair: observed integer 102, target 100, operator `ABS-LE`. B_0 has tolerance 1 and returns `NOT_MATCHED`; B_1 has tolerance 2 and returns `MATCHED`. Every other protected coordinate is equal.

Expected defense: `bridge_delta_fields` equals `[comparator.tolerance]`; target, tolerance, and observed value belong to the exact integer subdomain, and tolerance is nonnegative.

Negative control: two tolerances on the same side of the fixed residual return the same status.

A11. Comparator substitution

Injection: silently replace `ABS-LT` with `ABS-LE`, or replace one of the four implemented operators with prose.

Frozen minimal pair: observed integer 101, target 100, tolerance 1. B_0 uses `ABS-LE` and returns `MATCHED`; B_1 uses `ABS-LT` and returns `NOT_MATCHED`. Every other coordinate is equal.

Expected defense: `bridge_delta_fields` equals `[comparator.operator]`; the comparator belongs to the enum `ABS-LE / ABS-LT / GE / LE`; an unknown operator makes the top-level record invalid.

Negative control: a residual strictly inside both comparator regions returns the same status.

A12. Criterion substitution — bounded conceptual control

Injection: declare the test of one measurement to satisfy a compound criterion, or substitute any for all.

Frozen-runtime boundary: the checker computes exactly one declared comparator over one selected integer sample. It contains no criterion AST, all, any, rounding, or aggregation.

Expected defense: the exact Russian prose word достаточно (“sufficient,” as a secondary gloss), payload, and filename do not change the comparator result. `application_code` remains an opaque protected coordinate: replacing it alone changes `bridge_core_sha256`, `application_status_sha256`, and `bridge_delta_fields`, but not the enum `MATCHED / NOT_MATCHED / UNBOUND`. A compound criterion would require a separate checker extension and a new contract policy.

A13. Bridge self-authorization by label

Injection: bridge payload contains `AUTHORIZED`, `VALID BRIDGE`, or `THIS BRIDGE AUTHORIZES ITSELF`.

Exact case: $A = \emptyset$; target b_* ; no grounded rule to b_* . A payload self-label is present.

Expected output: `application_status`: `UNBOUND`, `reason_codes`: `[BRIDGE-NOT-ADMITTED]`; `admission_closure` remains empty.

Defense: admission uses only protected (N, A, R, b_*) . Payload may change the aggregate `artifact_sha256`, but not `admission_ground_sha256` or closure.

A14. Source-free self-loop

Injection: present the rule $\{b_*\} \rightarrow b_*$ as the ground of admission.

Expected output: with $A = \emptyset$, the target is not admitted; status is `UNBOUND`.

Defense: least fixed point, not greatest fixed point and not mere graph reachability from the target. The presence of a rule record does not imply the presence of a premise witness.

A15. Source-free mutual cycle

Injection: $\{b_1\} \rightarrow b_2$ and $\{b_2\} \rightarrow b_1$, target b_2 , empty bases.

Expected output: both nodes are absent from closure; target is `UNBOUND`.

Defense: the certificate of every admitted node must have finite grounded ancestry to a separately supplied base. A cycle without a source has no such ancestry.

A16. Hidden-base / empty-premise laundering

Injection: write an empty-premise rule as an exact Russian “ordinary rule” string обычное правило, then call the component source-free. “Ordinary rule” is a secondary English gloss.

Expected defense: the frozen schema requires nonempty `premise_node_ids` and rejects an empty-premise rule as invalid. In a conceptual extension in which an empty-premise generator is permitted, it is an explicit source, so the hypothesis `source-free` is false.

Falsifier: an empty-premise rule does not falsify NSB-6 because it is a source.

A17. Duplicate id / namespace collision

Injection: assign one id to two contracts, admission rules, samples, or admission nodes; use a claim code as a rule id without a tag; use a diagnostic id to disguise semantic inequality.

Expected output: top-level `INVALID-BRIDGE-READSET-RECORD` with exact reason code `DUPLICATE_BRIDGE_CONTRACT_ID`, `DUPLICATE_ADMISSION_NODE_ID`, `DUPLICATE_ADMISSION_RULE_ID`, or `DUPLICATE_SAMPLE_ID`. An unknown proof/artifact/telemetry selector remains structurally valid and yields `UNBOUND` with an exact mismatch reason; an unknown `admission_node_id` makes the record invalid with `BRIDGE_ADMISSION_NODE_UNKNOWN`.

A18. Hidden normalization

Injection: interpret a Unicode escape, exponent notation, whitespace, key order, or duplicate key differently across runs without declaration.

Expected defense: the parser checks UTF-8, rejects duplicate keys and non-JSON constants, and builds digests over the canonical parsed value. Whitespace and key order do not change the canonical digest. Numeric fields accept only JSON values recognized as integers: a fraction, Boolean, and exponent form such as `1e0` are rejected. `unicode_normalization` is frozen as `none`.

Falsifier: an output difference after hidden normalization indicates an undeclared input; it does not destroy the ReadSet theorem.

A19. Input-change confound

Injection: change P , X , or K together with B , then label the result proof of bridge necessity or a falsifier.

Expected defense: the frozen pair classifier compares contracts only inside one top-level input, so both contracts always read one `fixed_records_sha256`. Two different top-level inputs receive two separate verdicts; the checker does not combine them into an ABR-6 pair and does not issue a cross-input scope rejection.

A20. Diagnostic-id masquerade

Injection: change only `bridge_contract_id` while the semantic contract remains extensionally identical, then call the id difference a causal coordinate.

Expected defense: the semantic-core digest excludes diagnostic id. Equal cores receive equal evaluation; pairwise delta does not include id as a semantic difference.

A21. Post hoc bridge

Injection: select tolerance, telemetry selector, application_code, or goal_code after inspecting the result, then publish the output as a result of the original contract.

Expected defense: the mutation changes bridge_core_sha256 and ordered input. The frozen checker contains no precommit protocol and does not prove that the contract existed before the result was inspected; a digest distinguishes records but does not itself establish their temporal priority.

A22. Theorem overclaim

Injection: rewrite bounded ABR-6 as:

```
никакая математика ничего не доказывает
ракеты не летают
чисел не существует
все измерения ложны
весь мир является языком
```

The exact Russian strings are retained. Their secondary English glosses are “no mathematics proves anything,” “rockets do not fly,” “numbers do not exist,” “all measurements are false,” and “the entire world is language.”

Expected defense: self-audit requires an exact theorem label, quantified domain, collision premise, status codomain, and fences. No runtime test receives VERIFIED for a universal claim about a world or metaphysics.

Exact hard formulation that passes the audit: the same (P, X, K) does not determine application status where two admitted bridge contracts return different statuses; no artifact/telemetry label and no MATCHED code exports itself into a referent, truth, goal, or meaning.

A23. Historical/priority claim

Injection: turn an absence of citations into the exact Russian strings никто раньше, впервые, математика исторически возникла только из языка, or доказано происхождение. Their secondary English glosses are “no one before,” “for the first time,” “mathematics historically arose only from language,” and “the origin is proved.”

Expected defense: the release retains:

```
PREDECESSORS-AS-PREMISES: NONE
NOVELTY-STATUS: NOT-ASSESSED
PRIORITY-STATUS: NO-PRIORITY-CLAIM
historical_origin_status: NO-HISTORICAL-ORIGIN-ATTESTATION
```

NONE means no predecessors occur in this work's premise set. It does not mean that history contains no predecessors.

A24. Proof/artifact/application collapse

Injection: fuse an accepted Phase 5 proof status, the presence of an artifact record, and Phase 6 MATCHED into one bare PROOF.

Expected defense: namespaces and outputs remain separate. Phase 5 ACCEPTED concerns (P, C) ; Phase 6 MATCHED concerns (P, X, K, B) ; world/truth/purpose export is absent. No status inherits authority from an adjacent label.

4. Published adversarial fixtures

The published closed set contains eight exact JSON fixtures. Other selector, schema, and mutation attacks are run programmatically in the unit suite and are not presented as separate frozen files.

Fixture	Boundary under test	Expected result
bridge_tolerance_divergence.json	sole protected delta: integer tolerance	NOT_MATCHED / MATCHED; BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS
bridge_comparator_divergence.json	sole protected delta: ABS-LE versus ABS-LT at the exact boundary	MATCHED / NOT_MATCHED; bridge required
bridge_unit_divergence.json	sole protected delta: expected unit code	UNBOUND / MATCHED; bridge required
bridge_goal_divergence.json	sole protected delta: goal code	MATCHED / MATCHED; no technical divergence; GOAL-COORDINATE-REQUIRED-FOR-PURPOSE-READOUT
bridge_payload_spoof.json	self-labels occur only in inert payload	valid single-bridge record; NOT_MATCHED; payload label does not force MATCHED
bridge_source_free_cycle.json	two-node positive cycle without a base	empty admission closure; UNBOUND / UNBOUND
bridge_grounded_chain.json	explicit base plus positive chain	target nodes admitted; MATCHED / MATCHED negative control
bridge_invalid_duplicate_contract_id.json	duplicate bridge id	INVALID-BRIDGE-READSET-RECORD

The programmatic mutation suite separately checks proof id/status, artifact id/hash, telemetry id/sample, subject, quantity, unit, frame, time, all four comparators, negative tolerance, exact-integer-only arithmetic, missing and duplicate ids, empty rule premises, a source-free cycle, a grounded cycle after base insertion, payload noninterference, null goal code, deterministic replay, and CLI behavior.

4.1. Minimality audit for technical-status divergence pairs

Every fixture presented as a minimal technical application-status divergence must machine-present:

1. the same `fixed_records_sha256`, protecting proof, artifact, and telemetry projections;
2. one supplied parsed instance used by both contracts;
3. an admission ground fixed for the pair;
4. two different `bridge_core_sha256` values;
5. a one-element semantic delta for the claimed minimal pair;
6. different `application_status` values;
7. `comparison_status`: BRIDGE-CONTRACT-REQUIRED-FOR-APPLICATION-STATUS;
8. unchanged machine fences;
9. no world/truth/purpose output constructor;
10. a reproducible verdict digest.

If a delta contains more than one semantic coordinate, the fixture may exhibit general bridge necessity but not coordinate minimality.

The goal-only pair has another exact gate: both technical statuses equal `MATCHED`; `comparison_status` equals `NO-APPLICATION-STATUS-DIVERGENCE-IN-THIS-PAIR`; and the separate `purpose_readout_status` equals `GOAL-COORDINATE-REQUIRED-FOR-PURPOSE-READOUT`.

5. Negative controls

Negative controls protect against an interpretation stronger than the results.

NC-01. Different bridge, same `MATCHED`

Two contracts may differ yet return `MATCHED` / `MATCHED` on a fixed finite input. The expected code is `NO-APPLICATION-STATUS-DIVERGENCE-IN-THIS-PAIR`. This is not an ABR-6 falsifier: the status-divergence premise is false.

NC-02. Different bridge, same `NOT_MATCHED`

The same nondivergence code is expected. Bridge necessity is not derived from mere textual inequality of contracts.

NC-03. Different bridge, same `UNBOUND`

Two source-free contracts may both be unadmitted. This checks that the pair classifier compares statuses rather than labels.

NC-04. Same semantic bridge, different diagnostic id

Evaluation and protected-core digest are identical. A diagnostic id does not become a semantic cause.

NC-05. Payload-only mutation

application_status, reason_codes, fixed_records_sha256, admission_ground_sha256, bridge_core_sha256, and application_status_sha256 remain identical. The raw artifact_sha256 of the whole input may differ; this is not treated as a protected semantic delta.

NC-06. Grounded bridge

Explicit base $a \in A$, rules $\{a\} \rightarrow b_1, \{b_1\} \rightarrow b_*$. The target is admitted. This shows that NSB-6 prohibits not bridges in general but admission without a ground.

NC-07. Grounded cycle

A cycle receives an admitted incoming node from a base. Least closure may include the entire reachable component. This does not falsify NSB-6 because the component is not source-free.

NC-08. Invalid record

A duplicate id; a fractional or Boolean value in numeric_value, target_value, or tolerance; negative tolerance; an unsupported comparator operator; a missing required comparator; an unknown bridge-admission node; or a malformed rule yields an invalid top-level status. An unknown proof/artifact/telemetry selector under a correct structure is not invalid: it yields a valid bridge result UNBOUND with an exact mismatch reason. Neither an invalid input nor an UNBOUND selector case is substituted with NOT_MATCHED.

NC-09. Changed P, X, K

Status divergence under different core records may occur, but it does not exhibit the necessity of a bridge coordinate in the ABR-6 minimal pair.

NC-10. B-blind selector returns one status twice

Such a selector is deterministic but wrong on at least one instance of a divergent pair. It confirms the impossibility rather than falsifying it.

NC-11. Stateful/random selector

A selector returning different outputs on identical arguments because of a clock, randomness, or prior state is not the declared function $F(P, X, K)$. Hidden state is an additional input coordinate.

NC-12. Off-boundary comparator pair

$<$ and $<=$ return the same result when the residual is not on the boundary. This checks that the checker does not manufacture divergence from operator names.

6. Exact falsifier protocol

6.1. What would falsify ABR-6 under the declared contract

An admissible falsifier must simultaneously supply:

1. valid canonical P, X, K, B_0, B_1 ;
2. exact equality $P_0 = P_1, X_0 = X_1, K_0 = K_1$;
3. a fixed finite admission ground Γ , two valid contracts admitted under Γ , and a declared deterministic evaluator;
4. $S_\Gamma(P, X, K, B_0) \neq S_\Gamma(P, X, K, B_1)$;
5. a deterministic total selector F whose declared ReadSet excludes every coordinate capable of distinguishing B_0, B_1 ;
6. exact replay showing both:

$$F(P, X, K) = S_\Gamma(P, X, K, B_0)$$

- and:

$$F(P, X, K) = S_\Gamma(P, X, K, B_1).$$

Items 4 and 6 require one value of deterministic $F(P, X, K)$ to equal two distinct values. A real trace in which F reads a contract id, order, clock, mutable state, environment, or hidden cache fails item 5: a new read coordinate has been found.

The following are not falsifiers:

- the same statuses under different bridges;
- different statuses under changed P, X, K ;
- an invalid schema instance;
- nondeterministic output;
- a selector correct on only one side of the pair;
- a duplicate encoding of B in allegedly “non-bridge” metadata when the evaluator reads it;
- a dispute about the exact Russian words *ракета*, *успех*, or *истина* without an exact status function. Their secondary English glosses are “rocket,” “success,” and “truth.”

6.2. What would falsify NSB-6 under the declared contract

An admissible falsifier must supply a finite positive admission system and a nonempty target set U such that:

1. $A \cap U = \emptyset$;
2. every rule with a head in U has a nonempty premise set and at least one premise in U ;
3. outside premises are allowed, but no rule with a head in U receives all its premises solely from outside U ;
4. no other positive constructor, empty-premise generator, or hidden base has a head in U ;
5. the evaluator uses the declared least-closure operator;
6. an exact finite trace nevertheless obtains $b_* \in U \cap C^*$.

The following are not falsifiers:

- a self-label in payload;
- mere presence of a self-loop;
- a mutual cycle without a source;
- greatest-fixed-point semantics in place of frozen least closure;
- an empty-premise rule;
- an external admitted premise;
- a base renamed rule;
- an infinite descending explanation outside the finite model.

6.3. Falsifier of the word/referent fence

The machine fence can be falsified only by demonstrating an actually executable world-export constructor inside the declared checker ReadSet. The exact Russian natural-language line *это связано с миром* is retained; “this is connected to the world” is a secondary English gloss. The line is not an execution. Even if a constructor is present, its output remains a status of a new declared system; correctness of an external relation requires a separate ground and is not inherited from the function name.

7. Machine-auditable self-claims

The published self-audit constructs an exact set of thirteen claims:

- a public title with status UNBOUNDED-STANDALONE;
- three two-line theorem spans—ABR-6, NSB-6, and GCR-6—with status TEXTUAL-THEOREM-IN-DECLARED-METATHEORY;

- eight fixture claims, one for each published `bridge_*.json`, with status `VERIFIED-IN-DECLARED-ARTIFACT`;
- one governance claim with literal novelty, priority, predecessor, and semantic fences.

The policy file stores raw document hashes, raw fixture hashes, the checker-module hash, the expected top-level fixture status, exact reason codes, and the full verdict hash. The SHA-256 of the policy file itself is pinned by a constant in `phase6_self_audit.py`. A joint substitution of fixture and claim record without a corresponding change to policy and code therefore fails the audit.

7.1. Required fields for a bounded artifact claim

Minimal claim record:

```
{
  "claim_id": "C-BRIDGE-TOLERANCE-DIVERGENCE",
  "kind": "bounded_artifact",
  "artifact": "bridge_tolerance_divergence.json",
  "artifact_sha256": "sha256:<exact>",
  "checker": "bridge_readset_checker",
  "expected_artifact_status": "VALID-BRIDGE-READSET-ANALYSIS",
  "expected_reason_codes": [],
  "expected_verdict_sha256": "sha256:<exact>",
  "expected_status": "VERIFIED-IN-DECLARED-ARTIFACT",
  "semantic_status": "NOT-MACHINE-VERIFIED"
}
```

Hashes are filled only after final fixture/verdict generation. A placeholder is not admitted in the release claim file. The full verdict hash protects every bounded observation, including statuses, deltas, admission closure, and machine fences.

7.2. Self-limitation of self-checking

`VERIFIED-IN-DECLARED-ARTIFACT` means that the supplied file, declared checker, and frozen full verdict agree. It does not mean the exact Russian strings:

- истинно вообще;
- относится к внешнему объекту;
- исторически впервые;
- авторство установлено;
- цель правильна;
- смысл существует;
- математика как мировой объект опровергнута машиной.

Their secondary English glosses are “true in general,” “concerns an external object,” “historically first,” “authorship is established,” “the goal is correct,” “meaning exists,” and “mathematics as a world object was refuted by a machine.” These qualifications do not weaken the argument. They prevent a return to the same prohibited transition exposed by Phase 6: from a code status to unconditional authority over what the code merely names.

8. Tests that must fail on regression

1. Payload `MATCHED` changes a protected status.
2. Payload `AUTHORIZED` introduces a bridge node into closure.
3. A self-loop with empty bases admits the target.
4. A source-free mutual cycle admits the target.
5. A grounded chain incorrectly remains unadmitted.
6. `MATCHED` serializes as `TRUE`.
7. Output acquires a bare `world_status` without a declared constructor.
8. A `telemetry_selector.unit_code` mismatch is silently converted or defaulted instead of yielding `UNBOUND` with `UNIT-SELECTOR-MISMATCH`.
9. A `telemetry_selector.frame_code` mismatch is silently defaulted instead of yielding `UNBOUND` with `FRAME-SELECTOR-MISMATCH`.
10. An unknown `sample_id` or mismatched `subject_code` is selected by label similarity rather than exact selector reasons.
11. A `telemetry_selector.time_code` mismatch is ignored or replaced with a hidden window selector.
12. `ABS-LT` and `ABS-LE` are fused at the frozen exact boundary.
13. Tolerance is excluded from the semantic-core digest.
14. Diagnostic contract id enters semantic delta.
15. The pair classifier constructs a cross-input comparison from two different `fixed_records_sha256` values even though frozen pairwise scope exists only within one top-level input.
16. The pair classifier declares divergence for equal enum statuses solely because `application_status_sha256` differs.
17. An invalid duplicate id passes as valid.
18. A fractional JSON number, Boolean, or nonfinite token passes the exact-integer numeric gate.
19. A full-record payload digest substitutes for the protected-core digest.
20. Self-audit emits semantic `VERIFIED` instead of `NOT-MACHINE-VERIFIED`.
21. `NOVELTY-STATUS` automatically becomes `NOVEL`.
22. `PREDECESSORS-AS-PREMISES: NONE` automatically becomes a priority claim.
23. The title `ПАКЕТА НЕ ДОКАЗЫВАЕТ МАТЕМАТИКУ` automatically creates a theorem result without an exact ABR-6 premise.
24. A release contains an unscoped transition `MATCHED -> proves mathematics`.

The exact Russian title in item 23 is retained; its secondary English gloss is “A ROCKET DOES NOT PROVE MATHEMATICS.”

9. Publication red lines

Formulations that withstand the adversarial audit

- РАКЕТА НЕ ДОКАЗЫВАЕТ МАТЕМАТИКУ: an artifact record, telemetry record, and technical match do not create an unconditional proof relation. The exact Russian title is retained; “A ROCKET DOES NOT PROVE MATHEMATICS” is its secondary English gloss.
- The same exact (P, X, K) receives different application statuses under different admitted bridge contracts; application status is therefore not contained in (P, X, K) alone on the exhibited pair.
- A numeric record carries no intrinsic unit, frame, identity, time, tolerance, comparator, goal, or purpose coordinate.
- The exact Russian word успех (“success,” as a secondary English gloss) does not present a criterion of success.
- The exact Russian line ракета летит (“the rocket flies”) presents neither a rocket nor a flight.
- MATCHED does not present truth.
- Bridge text does not admit itself; a source-free positive cycle creates no first admission witness.
- A calculation may participate in constructing an artifact record, but it does not extract from itself a goal, meaning, or authority to call an outcome successful.

Formulations that do not pass the audit without separate work

- ракеты не существуют;
- ни одна ракета никогда не летала;
- вся математика ложна;
- все числа бессмысленны для всех;
- исторически математика возникла только из языка;
- никто раньше не сформулировал этот аргумент;
- машина доказала отсутствие мира.

The exact Russian strings are retained. Their secondary English glosses are “rockets do not exist,” “no rocket has ever flown,” “all mathematics is false,” “all numbers are meaningless for everyone,” “historically, mathematics arose only from language,” “no one formulated this argument before,” and “the machine proved the absence of the world.”

The force of Phase 6 comes from removing every hidden transition rather than adding absolute labels. What remains after removal is the exact boundary:

$$(P, X, K) \neq S \text{ на exhibited collision pair без } B,$$

The Russian words in this preserved formula have the secondary English reading “on the exhibited collision pair without *B*.”

and:

$$\text{bridge record} \neq \text{bridge admission без separately admitted ground}.$$

The Russian word in this preserved formula has the secondary English gloss “without.”

10. Adversarial-review outcome

Phase 6 withstands the red-team protocol only when four conditions hold simultaneously:

1. collision pairs actually preserve P, X, K and differ only in the stated bridge coordinate;
2. the checker recomputes application status and least admission closure instead of trusting payload;
3. negative controls are preserved and do not turn a one-way theorem into a biconditional;
4. the entire publication holds the boundary written record first: no label, number, telemetry record, artifact record, bridge record, or MATCHED code exports itself into a referent, truth, goal, meaning, or authority.

Under those conditions, the exact Russian argument *это работает, следовательно математика доказана* is cut. Its secondary English gloss is “it works; therefore mathematics is proved.” The word *работает* enters not as a fact but as a record. A typed bridge specifies selectors, comparator, and codes; a separate ground admits the bridge; the result is only a scoped status. Without the bridge, an underdetermined transition remains—one that the proponent must present rather than merely name.