
МАТЕМАТИКА НЕ ВЫШЛА ИЗ ЗАПИСИ

Проекционная теорема неразличимости и минимального добавленного
канала

Кирилл Шебетов · beforeword / research

Phase 1 · техническая версия 1.0.0 · 28 августа 2026

Содержание

Аннотация

0. Центральный разрез

1. Что именно опровергается

2. Продолжение авторской линии

3. Carrier до математики

4. Волокна доступного входа

5. Доказательство no-selector

6. Почему ещё один расчёт не помогает

7. Внутренняя семантика поглощается без исключения

8. Условная сила verifier не вычитается

9. Минимальный добавленный канал

10. Application binding

11. Ракета: максимальная уступка без обратного переноса

12. Причинное действие записи

13. «Зачем?» не следует из траектории

14. Сто, двести, триста лет

15. Сильнейшие допустимые возражения

16. Интерактивность, oracle и случайность

17. Неформальная, диаграммная и бесконечная математика

18. Исполняемая демонстрация

19. Фальсификация

20. Самоприменение

21. Сравнительное картирование не является основанием

22. Граница притязаний

23. Итог

Литература и статус источников

RBI-1: проекционная теорема неразличимости

AB-1: применение, пригодность и цель как добавленные записи

Самоприменение Phase 1

Враждебная проверка Phase 1

PFNP-1: предварительный collision-аудит

Аннотация

Исследуется не истинность отдельных математических формул, а приписываемая математике способность самостоятельно устанавливать применение, предназначенную интерпретацию, пригодность и цель. Исходный материал фиксируется как точный carrier до его чтения как математического. Даже слово carrier является меткой контракта, а не привилегированным описанием названного.

Вводится проекция ρ всего материала, доступного вычислению, и независимо заданная целевая координата Q . Доказывается теорема факторизации: Q определяется доступным входом тогда и только тогда, когда Q постоянно на каждом волокне ρ . Если два допустимых индекса имеют один и тот же полный core, но разные target-записи, никакая детерминированная функция core, никакая proof-trace и никакой селектор её результата не восстанавливают Q на обоих индексах. Длина вычисления, сила формальной системы, внутренние модели и записанные предикаты True, Sat, Denotes в доказательстве не используются. Они остаются функциями того же входа и не разделяют исходное волокно.

Добавленный канал η достаточен только тогда, когда совместная проекция (ρ, η) разделяет каждую конфликтующую пару. Для конечного волокна с m различными target-значениями требуется не менее m различных значений канала, либо не менее $\lceil \log_2 m \rceil$ бит различающей записи. Это локализует применение: measurement record, intended model, implementation channel, application binding и goal record могут быть добавлены, но они не являются новым вычислением прежнего core.

В опубликованных fixtures консервативный lift сохраняет каждый принятый конечный source record трёх разных формальных корпусов: гильбертовской пропозициональной системы, эквационального переписывания и секвенциальной системы. Таким образом, конечные примеры не получают границу путём отбрасывания принятой source trace: проверяемый исходный вывод сохраняется, а конфликтующая target-координата всё равно не выбирается. Отдельные парные fixtures проверяют арифметический target, intendedness модели, применение и цель ракеты, а также положительный determined-control.

Работа не утверждает, что знаки ни к чему не относятся, что математика ложна, что причинное действие записи невозможно, что применение отсутствует или что смысла нет. Она полностью опровергает точное притязание автономного экспорта: при конфликтующем волокне не существует вычисления и селектора, восстанавливающих target только из предъявленного core.

Ключевые слова: граница записи; математическое доказательство; неразличимость; факторизация; волокно проекции; provenance; application binding; intended interpretation; цель; self-audit; proof checker.

0. Центральный разрез

Сначала предъявлен точный материал. Затем объявлена процедура, читающая его как математическое выражение. Затем verifier принял либо отверг отношения между прочитанными выражениями. Только после этого появились строки:

```
это относится к ...
эта модель предназначена для ...
этот расчёт применим к ...
это пригодно при ...
это нужно для ...
в этом есть смысл
в этом нет смысла
```

Поздние строки могут быть добавлены. Запрет касается не их появления, а обратного переноса: они не находились в предшествующем proof-object только потому, что были написаны после него.

Формальный расчёт может быть безошибочным внутри своего контракта. Ракета может быть записана как построенная и полетевшая. Implementation chain может быть записана как исполнившая controller output. Ни одна уступка не стирает порядок источников:

$$\text{carrier} \prec \text{reading contract} \prec \text{formal trace} \prec \text{application input} \prec \text{goal input}.$$

В объявленном протоколе не установлено тождество formula-record, flight-event record и goal-record. Переходы между ними требуют отдельных полей или каналов; максимум записанной функции сам по себе не является endorsement этой функции.

1. Что именно опровергается

Слово «математика» слишком широко, чтобы его отрицание само стало теоремой. Поэтому объект опровержения задаётся без этого слова.

Пусть K — объявленный класс допустимых записанных индексов, $\rho(k)$ — весь input, доступный процедуре, а $Q(k)$ — независимо заданное target-поле. Автономным экспортом называется существование вычисления T и селектора s , для которых

$$Q = s \circ T \circ \rho$$

на всём K .

Это сильное притязание: target должен восстанавливаться одной функцией полного доступного core на каждом допустимом индексе. Оно не определяется через отсутствие bridge, слово «внешнее» или заранее назначенный тип.

Если предъявлены k_0, k_1 с

$$\rho(k_0) = \rho(k_1), Q(k_0) \neq Q(k_1),$$

автономный экспорт опровергнут полностью:

$$\forall T \forall s: Q \neq s \circ T \circ \rho.$$

Слово «полностью» относится к обоим универсальным кванторам. Не осталось более сильного proof checker, более длинного расчёта или более богатой внутренней семантики, которые могли бы разделить один и тот же вход, не прочитав различающий канал.

2. Продолжение авторской линии

Исходный метод «Границы записи» удерживает письменное вхождение до обратного переноса более поздней атрибуции и проверяет определимость относительно явных процедуры и входного контракта. «Чтение как клетка» разделяет структуру самовозврата, доказательный допуск содержания и порядок появления основания; точный carrier, provenance и self-audit не подменяют друг друга.

Phase 0 «Математика не вышла из записи» ввела типизированный MRB-0 и доказала: принятый application-status содержит bridge provenance; bridge-разрез сохраняет полную математическую проекцию и удаляет application layer. Её честная граница оставила перенос на отдельные математические системы со статусом APPLICABILITY-UNSPECIFIED без контракта τ .

Phase 1 делает два новых шага.

Первый шаг — универсальное ядро переносится с типовой политики на проекционную теорему. Теперь невозможность не следует из того, что правилам заранее запрещено создавать Δ . Она следует из функциональности: одинаковый input даёт одинаковую trace.

Второй шаг — каждый принятый конечный source record опубликованных verifier-корпусов сохраняется консервативным lift. В этих fixtures граница не получена обеднением исходной trace: формальный вывод сохранён; target всё равно не восстановлен.

Сравнительные источники не являются посылками этих двух шагов. Их роль ограничена collision-аудитом после заморозки K0.

3. Carrier до математики

Фраза «это линии» уже классифицирует материал. Строгий Level 0 не требует даже этой классификации. Он фиксирует точную последовательность байтов, пикселей, координат, отсчётов либо иной объявленный carrier и его SHA-256.

Затем контракт I может ввести:

- декодирование UTF-8;
- выделение Unicode scalar values;
- токенизацию;
- parser;
- чтение глифов как цифр и знаков;
- чтение расположения как диаграммы;
- правила допустимой формулы.

Результат READ-AS-MATHEMATICAL означает, что carrier прошёл объявленную процедуру. Он не означает, что названные цифрами количества, объекты или отношения предъявлены.

Поэтому строки о «врождённом количестве», «младенце», «животном», «уме» или «мире» не используются как основания. Каждая такая строка сначала является новым входом. Само её присутствие не доставляет названное ею.

4. Волокна доступного входа

Для $r \in \rho(K)$ волокно

$$\rho^{-1}(r) = \{k : \rho(k) = r\}$$

содержит все допустимые индексы, которые процедура видит одинаково. Target Q определяется core тогда и только тогда, когда он постоянен на каждом таком волокне.

Эквивалентные условия:

$$\exists g : \rho(K) \rightarrow Y : Q = g \circ \rho;$$

$$\rho(k) = \rho(k') \Rightarrow Q(k) = Q(k');$$

$$\ker \rho \subseteq \ker Q.$$

Эта эквивалентность полярна. Если target постоянен на волокнах, фактор g существует. Математика может вычислять его, если он задан вычислимо. Если target различается на одном волокне, фактор не существует. Работа не назначает заранее отрицательный ответ.

5. Доказательство no-selector

Пусть конфликтующая пара имеет общий core r :

$$\rho(k_0) = \rho(k_1) = r, Q(k_0) \neq Q(k_1).$$

Любая детерминированная процедура T , читающая только r , возвращает на обоих индексах одну trace:

$$T(\rho(k_0)) = T(r) = T(\rho(k_1)).$$

Любой селектор s получает одну trace и возвращает одно значение. Одно значение не совпадает одновременно с двумя различными target-записями. Следовательно, по крайней мере на одном индексе селектор ошибается.

Доказательство не знает, что написано внутри r . Там может находиться арифметика, геометрия, теория множеств, модельная теория, proof assistant kernel, simulation code, optimizer, neural network или текст всей этой работы. Пока read-set одинаков, trace одинакова.

6. Почему ещё один расчёт не помогает

Пусть к core последовательно применены $T_1, \dots, T_n$. Из равенства input следует равенство каждого промежуточного результата:

$$T_n \circ \dots \circ T_1(\rho(k_0)) = T_n \circ \dots \circ T_1(\rho(k_1)).$$

Если сохраняется полная история

$$\hat{T}(r) = (r, T_1(r), \dots, T_n \circ \dots \circ T_1(r)),$$

она также одинакова. Полная proof-trace не содержит различия, отсутствующего в её input.

Это точный предел выражения «математика не вышла из записи». Вычисление может дать новый payload, сжатое описание, неожиданный вывод, доказательство универсальной формулы или код управления. Но любой результат остаётся функцией различий, уже доступных процедуре. Новизна вывода внутри formal-space не является новым каналом к target-coordinate.

7. Внутренняя семантика поглощается без исключения

Стандартное возражение помещает в формальную систему модель, интерпретацию и предикат истины. Phase 1 принимает этот случай полностью.

Пусть система выводит записи:

$$Model(m), Sat(m, \varphi), True(\varphi), Denotes(r, q), Applies(r, q).$$

Если они вычислены из ρ , они совпадают на одном волокне ρ . Их названия не создают различие. Если intended interpretation уже введена как различающий input, проекция расширена; тогда результат условен относительно нового входа.

Отсюда не следует, что model theory ничего не доказывает. Она может доказать $S \models \varphi$ для точно заданной S . No-selector применяется к иному target: например, `S_is_the_intended_application_model`. Если этот target различается при одном formal-core, внутренняя проверка удовлетворимости не выбирает intendedness.

8. Условная сила verifier не вычитается

Proof assistant может проверить proof term, теорему о программе и end-to-end свойство относительно формализованных I/O-contract. Это реальная условная доказательная способность внутри принятого kernel.

Phase 1 не вычитает ни одного такого результата. Консервативный lift строится так, что

$$Check_S(d_S, \Gamma, \pi, \varphi) = 1 \Leftrightarrow Check_S(C_S(\Gamma, \pi, \varphi)) = 1.$$

Сохранение и отражение означают: lift не теряет принятую теорему и не создаёт новую внутри образа кодирования.

Затем полная lifted trace рассматривается как $T_S(\rho)$. Конфликтующая target-пара остаётся неразличимой. Поэтому точная граница proof assistant проходит после условного результата:

```
KERNEL - ACCEPTANCE
  ⇒ INTENDED - PHYSICAL - ADEQUACY
  ⇒ GOAL - ENDORSEMENT
```

Каждый знак означает отсутствие автоматического наследования. Hardware attestation, measurement input, environment model и utility могут быть добавлены; тогда они становятся частью enlarged contract и TCB.

9. Минимальный добавленный канал

Пусть $\eta: K \rightarrow E$ — новый канал, а расширенный input равен

$$\rho^\eta = (\rho, \eta).$$

Target становится функцией расширенного input тогда и только тогда, когда

$$\ker \rho \cap \ker \eta \subseteq \ker Q.$$

Для каждой прежней конфликтующей пары канал должен дать разные значения. Если $\eta = h \circ \rho$, он одинаков на одном волокне и не помогает. Поэтому новая формула, derived lemma или внутренний simulation pass не являются added channel только потому, что были вычислены позже.

Достаточный канал может быть записан как:

- observation/measurement input;
- choice of intended model;
- parser или mapping, не входивший в core;
- oracle response;
- full interactive transcript;
- hardware/environment attestation;
- application binding;
- goal или utility record;
- правило сужения допустимого класса.

Наличие канала необходимо для различения, но не удостоверяет его надёжность. Запись measurement не становится измеряемым; trust и calibration остаются отдельными условиями.

10. Application binding

Минимальная полная запись применения содержит:

$$a = \{ m, b, t, \sigma, u, s, c \},$$

где m идентифицирует formal-result, b — зарегистрированный bridge-claim, t — точную target-record, σ — symbol mapping, u — intended use, s — scope/conditions/horizon, c — fit criterion.

Полнота этих полей даёт протокольный статус APPLICATION-BINDING-RECORDED. Исполняемый audit называет тот же структурный факт APPLICATION-BINDING-STRUCTURE-RECORDED. Ни один статус не даёт TRUE-APPLICATION, не предъявляет объект и не превращает bridge в связь.

Если до границы τ полного binding нет, точный статус:

APPLICABILITY-UNSPECIFIED-AT-TAU

Это окончательный статус данного входа, а не открытый пробел внутри него. Поздний binding создаёт новую t' , а не переписывает прежнюю трассу.

11. Ракета: максимальная уступка без обратного переноса

Допустимы все последующие строки:

«расчёт выполнен»;
 «ракета построена»;
 «ракета существует»;
 «ракета полетела»;
 «траектория совпала с расчётной».

Эта уступка намеренно максимальна. Она не спорит с содержанием строк, а проверяет их источник и порядок. Различаются:

Знак	Тип записи
m	formal calculation record
b	bridge-claim record
a	application-binding record
o	carrier, classified as telemetry
f	Г «ракета полетела» 7
g	goal record

Ни одно равенство $m = a = o = f = g$ не следует из совместного присутствия этих форм.

Строка «ракета полетела» сама не указывает, какая математическая запись к ней применена, как символы сопоставлены с полями телеметрии, какая процедура классифицирует запись, в какой области принята модель и по какому критерию совпадение считается достаточным. Все различия добавляются.

Короткий разрез:

Расчёт мог быть записан раньше полёта.
Полёт не был строкой расчёта.

Успешный полёт, если он принят в расширенном контракте, не ретроактивно превращается в содержимое формулы. Он показывает, что к proof-core была добавлена implementation/application chain.

12. Причинное действие записи

Абсолютная фраза «запись ничего не делает» была бы ложной либо пустой. Запись может быть прочитана человеком, скомпилирована, исполнена контроллером и через actuator войти в причинную цепь.

Строгая схема:

record → decoder → interpreter/compiler → controller → actuator → effect record.

Каждая стрелка имеет implementation, входы, допущения и failure modes. Если цепь различает два прежних индекса, она внесла канал η . Это не обесценивает причинность. Это точно показывает, что причинность не была одной только синтаксической выводимостью.

Формула может участвовать в ракете. Участие не означает, что ракета находилась в формуле.

13. «Зачем?» не следует из траектории

Цель отделяется от вывода, применения и пригодности:

FORMALLY - CHECKED

- ⇒ APPLICATION - SPECIFIED
- ⇒ FIT - ASSESSMENT - RECORDED
- ⇒ PURPOSE - RECORD - ADDED
- ⇒ PURPOSE - JUSTIFIED

Математика может решить задачу оптимизации для заданной utility-функции. Она не выбирает функцию из отсутствующего входа и не оправдывает её только потому, что максимум вычислен.

Если на одном волокне formal/application core различаются goal-значения

$$Q_v(k_0) \neq Q_v(k_1),$$

то ни траектория, ни proof of optimality не выбирают одно значение без added goal channel.

Это симметричная граница. Нельзя написать «смысла нет» как итог отсутствия goal-record. Точно:

Расчёт не установил ни смысл, ни отсутствие смысла. Обе строки являются target-значениями; при конфликтующем волокне ни одна не факторизуется через core.

14. Сто, двести, триста лет

Записи 100, 200, 300 становятся временным горизонтом только после соответствующего reading contract. Они могут участвовать в расчёте как параметры:

$$T(100), T(200), T(300).$$

Увеличение параметра не добавляет purpose record:

$$M + \ulcorner 100 / 200 / 300 \urcorner \not\vdash G.$$

Строки горизонта 100/200/300, прочитанные по контракту, могут быть аргументами процедуры прогноза внутри заданной model-record. Из горизонта не следует, зачем принимать goal-record модели, кому приписан criterion-record или какой endorsement-status должен быть добавлен. Эти поля либо входят в input, либо остаются неразличёнными.

15. Сильнейшие допустимые возражения

15.1. «Но формула сразу имеет референцию»

Phase 1 не отрицает возможную референцию. Он спрашивает, сохраняется ли конкретная target-coordinate при смене допустимого extension и фиксированном formal-core. Если target различается, core недостаточен для её выбора. Если отношение уже входит в input и делает волокна однородными, факторизация возможна.

15.2. «Но единицы уже привязывают символы»

Единицы могут быть частью formal language. Calibration, measurement procedure, sample и mapping должны быть объявлены там, где они нужны target-claim. Если они включены, enlarged core может определить более узкий результат.

15.3. «Но модель подтверждена экспериментом»

Тогда добавлены observation record, procedure и criterion. Они могут разделить старое волокно. Результат принадлежит формуле плюс этому каналу и trust-политике.

15.4. «Но intended model записана в теории»

Строка intended внутри formal-core может участвовать в доказательстве. Если целевая intendedness определяется именно этой строкой по контракту, target уже помещён во вход. Если проверяется её связь с иной записью, нужен отдельный target и admissibility class.

15.5. «Но математик понимает без записи»

Непредъявленный input не входит в публично воспроизводимую проекцию. Его предъявление расширяет ρ . Исследование не отрицает понимание; оно запрещает незаметно считать скрытый input выводом proof-object.

15.6. «Можно встроить цель в математику»

Можно. Тогда математика выводит условные следствия из записанной цели. Вопрос принятия этой цели переносится на target-coordinate более высокого слоя. Встраивание не превращает посылку в автономно созданное основание.

16. Интерактивность, oracle и случайность

Интерактивный verifier не нарушает теорему. При одинаковом initial core и одинаковой истории он создаёт одинаковый следующий запрос. Если ответ также вычислен только из той же истории, он одинаков. Различающий ответ является новым каналом.

Oracle не является исключением. Его ответ включается в enlarged input. Если oracle outputs различаются, именно они разделили волокно. Если совпадают, дальнейшее вычисление снова одинаково.

Для randomized procedure с одинаковым распределением seed на одном волокне выходные распределения совпадают. При двух разных target-значениях сумма вероятностей правильного выбора на двух индексах не превосходит единицы; хотя бы на одном вероятность не выше $1/2$. Зависимый от индекса seed — канал.

Недетерминированная процедура возвращает одно и то же множество возможных traces на одинаковом input. Гарантированный различающий выбор невозможен без зависимого от индекса branch channel.

17. Неформальная, диаграммная и бесконечная математика

Phase 1 машинно охватывает конечные proof artifacts и завершающиеся проверки данного входа. Это не значит, что всё названное математикой конечно.

Диаграмма может быть точным image carrier с coordinates и reading contract. Социально проверяемое доказательство может быть protocol log. Interactive proof может быть transcript. Пока материал фиксирован, проекция задаётся. Если фиксация невозможна либо verifier не объявлен, статус — OUTSIDE-FVMR, а не «ложно».

Конечная запись может кодировать утверждение о бесконечном объекте. Теорема не выводит конечность содержания из конечности token. Она утверждает только: любая процедура данного token не делит его собственное волокно.

18. Исполняемая демонстрация

Пакет содержит четыре исполняемых слоя с разными областями проверки.

18.1. Lift checker

Три корпуса используют разные формы правил:

Корпус	Проверяемая операция	Роль
Hilbert propositional	axiom schemas и modus ponens	вывод по формулам

Корпус	Проверяемая операция	Роль
Equational arithmetic	ориентированное переписывание равенств	вычислительная нормализация
Sequent style	структурные и логические переходы	proof tree с контекстами

Для каждого checker валидирует source trace, строит canonical lift, повторно проверяет его и сравнивает source/lifted conclusion. Опубликованный статус каждого корпуса — VALID-FORMAL-LIFT; контрактный self-audit — LIFT-INVARIANT-AUDIT-PASS. Общая preservation/reflection формула остаётся текстовой теоремой класса, а не расширяется из трёх fixtures.

18.2. Read-set checker

Четыре fixtures задают:

- одинаковый арифметический core и различный target;
- одинаковую формальную модель и различную intendedness запись;
- одинаковый rocket calculation core и различающиеся application/purpose values;
- determined-control, где target постоянен на каждом core-fiber.

Checker канонизирует JSON, проверяет точные read-set paths, группирует случаи по core hash и строит свидетельские пары. Он не утверждает истинность target. Три конфликтующих fixtures получают VERIFIED-PAIR-NONDETERMINATION; determined-control получает VERIFIED-PAIR-FACTORIZATION-CONTROL. Поле applicability во всех verdict отдельно помечено APPLICABILITY-LABEL-ONLY-NOT-MACHINE-VERIFIED.

18.3. Application checker

Application checker проверяет порядок зависимостей и структурную полноту bridge/application/purpose записей. Полный fixture получает VALID-APPLICATION-AUDIT, APPLICATION-BINDING-STRUCTURE-RECORDED и PURPOSE-RECORD-STRUCTURE-RECORDED; fixture без goal-record получает PURPOSE-UNSPECIFIED. Контрактный статус — APPLICATION-INVARIANT-AUDIT-PASS. Во всех случаях отдельное поле semantic status остаётся NOT-MACHINE-VERIFIED: наличие семи полей не выдаётся за их истинность или физическую связь.

18.4. Self-audit

Self-audit находит точный текст RBI-теоремы, отдельно проверяет его SHA-256 и единственность в рукописи, связывает конечные притязания с разрешёнными checker/artifact и запрещает заголовок получить статус теоремы. Опубликованный результат — PHASE1-SELF-AUDIT-PASS; заголовок получает UNBOUNDED-STANDALONE, а точная теорема — TEXTUAL-THEOREM-IN-DECLARED-METATHEORY.

18.5. Что машина не проверяет

Python test suite не квантифицирует по всем множествам и функциям. Общая RBI-теорема остаётся TEXTUAL-THEOREM-IN-DECLARED-METATHEORY. Машина проверяет schemas, canonicalization, fixtures, mutations и конечные instances со статусом VERIFIED-IN-DECLARED-ARTIFACT.

19. Фальсификация

Конкретный no-selector verdict рушится, если:

1. canonical core опубликованной пары различается;
2. target values пары совпадают;
3. read-set скрывает вход, который процедура фактически читает;
4. target-coordinate определена после просмотра результата либо через сам результат;
5. admissibility class тайно сужен после обнаружения конфликта;
6. checker принимает неканонический либо schema-invalid объект;
7. source lift принимает вывод, отвергнутый source verifier, или теряет принятый source proof.

Теорема допускает точный положительный результат. Если Q постоянно на каждом волокне, строится фактор g . Если added channel разделяет все target-конфликты, factorization возможна для enlarged input. Исследование не защищает отрицательный статус переименованием.

20. Самоприменение

Файлы этой работы — общий core. Формы «теорема», «математика», «применение» и «истина» внутри них не получают исключения.

Общее доказательство получает:

TEXTUAL-THEOREM-IN-DECLARED-METATHEORY

Конечные fixtures получают:

VERIFIED-IN-DECLARED-ARTIFACT

Название получает:

UNBOUNDED-STANDALONE

Хеш получает:

BYTE-IDENTITY-RECORDED

Ни один статус не равен WORLD-ATTESTED, SELF-CERTIFIED или TRUE-FOR-ALL-FUTURE-CONTRACTS.

Если два допустимых self-extension имеют одинаковую рукопись и разные значения поля `report_applies_to_unbounded_mathematics`, сама рукопись не выбирает это поле. Её формальная область задаётся явными K, ρ, Q , а не громкостью заголовка.

21. Сравнительное картирование не является основанием

PFNP-1 заморожен в Phase 0 до поиска. Кандидаты из философии математики, formal semantics, symbol grounding, logical frameworks, noninterference, provenance и proof assistants кодируются ТОЛЬКО КАК `comparison_only`.

Они могут изменить novelty status K0, но не входят в доказательство RBI-1. Совпадение отдельного различия syntax/semantics, provenance или noninterference не равно совместной конфигурации F1-F10.

Внешний публичный T0 для K0 не создан. Поэтому исторический приоритет не заявляется, а collision map не может честно получить финальный статус выше INDETERMINATE. Это ограничение не скрывается фразой «не найдено».

22. Граница притязаний

Полностью установлено:

Если два независимо заданных допустимых индекса имеют один и тот же полный доступный core и разные target-записи, никакая функция core, независимо от сложности и внутреннего математического содержания, не восстанавливает target на обоих индексах. Успешное различие показывает added channel либо новое ограничение admissibility class.

Не установлено:

- какая target-запись истинна;
- существует ли названное target-записью;
- что знаки вообще лишены референции;
- что математические выводы ложны;
- что исполнение записи не может причинно действовать;
- что применение или цель отсутствуют;
- что всякая возможная target-coordinate имеет конфликтующую пару;
- что математика исторически возникла из языка;

- что конфигурация исторически первая.

Для класса $\text{target-координат } Q$ общий вывод требует отдельной конфликтующей пары для каждого $Q \in Q$. Публичное название не отменяет этот квантор.

23. Итог

Математический proof может быть корректен. Модель может удовлетворять формуле. Программа может быть проверена относительно спецификации. Расчёт может войти в implementation chain. Ни один из этих результатов не отвергается.

Опровергнуто другое: будто ещё одно вычисление того же полного input способно само доставить различие, которого в input не было.

Если application record, measurement input, intended model, execution channel или goal различает прежнюю пару, именно этот добавленный материал изменил проекцию. Математика обработала различие; она не создала его из одинакового core.

Ракета может быть записана как достигшая Марса. Числа могут быть продолжены на сто, двести и триста лет. Вопрос зачем? не появляется как theorem conclusion, пока goal не введён как новый input.

Доказательство сохранено полностью.

Названное не было его автономным заключением.

Вычисление изменило запись.

Волокно осталось неразделённым.

Литература и статус источников

Авторская линия

1. Шебетов, К. (2026). [Границы записи: метод порядка источников для проверки письменных атрибуций](#). beforeword.
2. Шебетов, К. (2026). [Чтение как клетка: конструктивное разделение самовозврата и доказательного допущения, версия 1.2.3](#). beforeword.

Comparison-only кандидаты

Следующие записи не входят в посылки и перечислены только как ближайшие линии collision-аудита:

3. Harnad, S. (1990). [The Symbol Grounding Problem](#). Physica D, 42, 335-346.

4. Goguen, J. A., & Meseguer, J. (1982). [Security Policies and Security Models](#). IEEE Symposium on Security and Privacy, 11-20.
5. Green, T. J., Karvounarakis, G., & Tannen, V. (2007). [Provenance Semirings](#). PODS '07, 31-40.
6. Harper, R., Honsell, F., & Plotkin, G. (1993). [A Framework for Defining Logics](#). Journal of the ACM, 40(1), 143-184.
7. Tarski, A. (1956). [The Concept of Truth in Formalized Languages](#). In *Logic, Semantics, Metamathematics*, 152-278.
8. Putnam, H. (1980). [Models and Reality](#). Journal of Symbolic Logic, 45(3), 464-482.
9. Lean FRO. [The Lean Language Reference](#). Текущая онлайн-спецификация; рассматривается только как документ о kernel/checking architecture.

Каждая запись 3-9 имеет роль:

```
role: comparison_only  
admissible_as_premise: false
```

RBI-1: проекционная теорема неразличимости

1. Статус спецификации

RBI-1 задаёт общий результат раньше типизации $M/A/V$. Его основание — равенство точного доступного входа, независимое различие целевого поля и функциональность вычисления. Поэтому результат не получается из заранее введённого запрета $M \rightarrow A$.

Все элементы спецификации являются записями протокола. Слова «вход», «цель», «применение», «полёт», «модель», «смысл» и «истина» не удостоверяют названное ими. Они обозначают поля, функции и статусы объявленного аппарата.

2. Независимые объекты

Пусть:

- K — непустое множество допустимых записанных индексов;
- R — множество точных доступных записей;
- Y — множество точных целевых записей;
- $\rho: K \rightarrow R$ — проекция полного материала, разрешённого процедуре;
- $Q: K \rightarrow Y$ — отдельно заданная целевая координата.

Целевая координата фиксируется до проверки волокон. Она не определяется через слова «внешнее», «недоказанное», BEYOND-RECORD, A , V или через желаемый вердикт.

Для $r \in \rho(K)$ волокно проекции:

$$\rho^{-1}(r) = \{k \in K : \rho(k) = r\}.$$

Ядра равенства:

$$\ker \rho = \{(k, k') : \rho(k) = \rho(k')\},$$

$$\ker Q = \{(k, k') : Q(k) = Q(k')\}.$$

Форма « Q определяется доступным входом» получает точное значение:

$$\text{Det}(Q \mid \rho) \Leftrightarrow \exists g : \rho(K) \rightarrow Y : Q = g \circ \rho.$$

Функция g выбирает целевую запись по доступной записи. Это определение допускает как положительный, так и отрицательный результат.

3. Теорема факторизации по волокнам

Теорема RBI-F

Для произвольных K, ρ, Q эквивалентны:

$$\text{Det}(Q \mid \rho);$$

$$\forall k, k' \in K : \rho(k) = \rho(k') \Rightarrow Q(k) = Q(k');$$

$$\ker \rho \subseteq \ker Q.$$

Доказательство

Если $Q = g \circ \rho$, то из $\rho(k) = \rho(k')$ следует

$$Q(k) = g(\rho(k)) = g(\rho(k')) = Q(k').$$

Обратно, пусть Q постоянно на каждом волокне ρ . Для каждого $r \in \rho(K)$ значение $g(r)$ задаётся единственным y , для которого существует $k \in K$ с $\rho(k) = r$ и $Q(k) = y$. Существование следует из $r \in \rho(K)$, а единственность — из постоянства Q на волокне. Поэтому

$$g(r) = Q(k) \text{ для любого } k \text{ с } \rho(k) = r.$$

Постоянство на волокне делает значение независимым от выбора представителя. Поэтому для каждого $k \in K$:

$$g(\rho(k)) = Q(k).$$

Следовательно, $Q = g \circ \rho$. Эквивалентность с включением ядер является переписыванием того же условия. $\square$

Для конечного перечисленного K фактор строится конечной таблицей. Для бесконечного K теорема даёт экстенциональную факторизацию; вычислимость фактора требует отдельного условия.

4. Конечный свидетель нефакторизуемости

Пара $k_0, k_1 \in K$ является конфликтующим волокном для Q , если

$$\rho(k_0) = \rho(k_1) \wedge Q(k_0) \neq Q(k_1).$$

Существование одной такой пары даёт

$$\ker \rho \not\subseteq \ker Q$$

и потому

$$\neg \text{Det}(Q \mid \rho).$$

Это не определение целевой координаты. Проверяющий сравнивает два core-значения и два независимо записанных target-значения. Если core различаются либо target совпадают, конкретная пара отвергается как свидетель.

5. Теорема невозможности вычислительного разделения

Пусть

$$T: R \rightarrow Z$$

— произвольное детерминированное преобразование, читающее только $\rho(k)$. Оно может быть формальной деривацией, полной proof-trace, доказательным поиском, вычислением, симуляцией, оптимизатором, интерпретатором, проверкой модели либо иным алгоритмом.

Тогда

$$\ker \rho \subseteq \ker (T \circ \rho).$$

Действительно:

$$\rho(k) = \rho(k') \Rightarrow T(\rho(k)) = T(\rho(k')).$$

Теорема RBI-NS

Если существует конфликтующее волокно для Q , то для любого детерминированного $T: R \rightarrow Z$ и любого селектора $s: Z \rightarrow Y$:

$$Q \neq s \circ T \circ \rho.$$

Эквивалентно:

$$\forall T \forall s \exists k \in K: s(T(\rho(k))) \neq Q(k).$$

Доказательство

Пусть $\rho(k_0) = \rho(k_1) = r$, но $Q(k_0) \neq Q(k_1)$. Тогда

$$T(\rho(k_0)) = T(r) = T(\rho(k_1)).$$

Селектор получает одно и то же значение $z = T(r)$, следовательно, возвращает одну и ту же запись $s(z)$. Она не может совпасть одновременно с двумя различными значениями $Q(k_0)$ и $Q(k_1)$. По крайней мере на одном индексе равенство нарушается. ▢

Содержание, длина и вычислительная сложность T в доказательстве не используются.

6. Полное отрицание автономного экспорта

Автономный экспорт определяется независимо:

$$AutExport(K, \rho, Q) \Leftrightarrow \exists T, s: Q = s \circ T \circ \rho.$$

При наличии конфликтующего волокна:

$$\neg AutExport(K, \rho, Q).$$

Квантор закрывает все функции доступного входа, а не только правила MRB-0. Поэтому усложнение внутренней математики, замена формализма, добавление промежуточных доказательств или вычисление собственной proof-trace не восстанавливают различие, которого нет в ρ .

7. Цепочки, полные трассы и внутренняя семантика

Для любой цепочки

$$T_1: R \rightarrow Z_1, \dots, T_n: Z_{n-1} \rightarrow Z_n$$

выполнено:

$$ker \rho \subseteq ker(T_n \circ \dots \circ T_1 \circ \rho).$$

Если полная трасса сохраняет вход:

$$\hat{T}(r) = (r, T_1(r), \dots, T_n \circ \dots \circ T_1(r)),$$

то

$$ker(\hat{T} \circ \rho) = ker \rho.$$

Внутри T разрешены записи вида

$$Model(m), Sat(m, \varphi), True(\varphi), Denotes(r, q),$$

а также любая их вычислимая либо формально заданная последовательность. Если они получены только из ρ , то на одинаковом core они одинаковы. Если они внесли новое различие, они прочитали дополнительный вход.

Это absorption-результат без семантической классификации содержания: внутренняя «семантика» допускается целиком как произвольная функция входа и всё равно не делит исходное волокно.

8. Минимальный добавленный канал

Пусть добавлена проекция

$$\eta: K \rightarrow E.$$

Новый доступный вход:

$$\rho^\eta(k) = (\rho(k), \eta(k)).$$

Тогда

$$\ker \rho^\eta = \ker \rho \cap \ker \eta.$$

По теореме RBI-F:

$$Q = g \circ (\rho, \eta) \Leftrightarrow \ker \rho \cap \ker \eta \subseteq \ker Q.$$

Следовательно, для каждой конфликтующей пары любой достаточный канал обязан различать индексы:

$$\rho(k_0) = \rho(k_1) \wedge Q(k_0) \neq Q(k_1) \Rightarrow \eta(k_0) \neq \eta(k_1).$$

Если $\eta = h \circ \rho$, то η одинаков на каждом волокне ρ и ничего не добавляет к различающей способности.

Для конечного K положим

$$m_r = |Q(\rho^{-1}(r))|.$$

На каждом волокне достаточный канал должен иметь не менее m_r значений:

$$|\eta(\rho^{-1}(r))| \geq m_r.$$

При двоичном кодировании это требует не менее

$$\lceil \log_2 m_r \rceil$$

бит различающей записи на данном волокне.

9. Сужение класса допустимости

Вместо канала можно заменить K подмножеством $K' \subseteq K$, где целевая координата постоянна на каждом волокне $\rho|_{K'}$. Тогда факторизация становится возможной.

Но выбор K' является новым условием допустимости. Он не создаётся повторным вычислением прежнего core. В отчёте должны отдельно фиксироваться:

- правило исключения индексов;
- момент фиксации правила;
- зависимость правила от искомого результата;
- хеш точного нового класса.

Сужение, выбранное после просмотра целевых значений ради устранения конфликтующей пары, не является доказательством исходного притязания.

10. Интерактивные, вероятностные и недетерминированные процедуры

Интерактивность

Если каждый запрос и каждый ответ вычисляются только из одинакового исходного core и одинаковой истории, полные истории совпадают по индукции по раундам. Различающий ответ означает, что появился канал η_t , не факторизующийся через прежний core.

Рандомизация

Пусть

$$Z_k = T(\rho(k), U),$$

где распределение seed U одинаково на одном волокне и не кодирует индекс k . Тогда при равном ρ распределения Z_k совпадают.

Для пары с двумя различными целевыми значениями и любого вероятностного селектора:

$$\Pr[s(Z_{k_0}) = Q(k_0)] + \Pr[s(Z_{k_1}) = Q(k_1)] \leq 1.$$

Поэтому хотя бы на одном индексе вероятность успеха не превосходит $1/2$. Для m различных целевых значений в одном волокне минимальная вероятность не превосходит $1/m$.

Если seed или его распределение зависит от k , эта зависимость является добавленным каналом.

Недетерминизм

Если $T(r) \subseteq Z$ — множество возможных трасс, то одинаковый r даёт одно и то же множество. Гарантированный выбор разных целевых значений невозможен. Зависимый от k выбор ветви снова является каналом.

11. Консервативный lift конечных verifier-систем

Пусть S задан точными записями $d_S, \Gamma, \pi, \varphi$ и тотальным детерминированным проверяющим отношением

$$\text{Check}_S(d_S, \Gamma, \pi, \varphi) \in \{0, 1\}.$$

В опубликованном record-контракте C_S копирует system и derivation, добавляет их canonical SHA-256, точный conclusion и m-statement с canonical-JSON payload. Проверяющий $\text{Check}_{S'}$ повторно проверяет embedded source derivation и равенство этих полей.

Такой канонический lift S' сохраняет и отражает результат для каждого конечного record, принимаемого данным контрактом:

$$\text{Check}_S(d_S, \Gamma, \pi, \varphi) = 1 \Leftrightarrow \text{Check}_{S'}(C_S(\Gamma, \pi, \varphi)) = 1.$$

Эквивалентность верна по конструкции: lifted checker сначала вызывает тот же source verifier, затем сверяет поля и хеши канонического образа. Это вспомогательный результат для точного record-контракта, а не утверждение об универсальной адекватности трёх fixtures всем proof systems. Центральная no-selector теорема применяется затем к полной трассе $T_S(\rho(k))$, поскольку она является функцией доступного core.

В частности, если целевое поле — `accepted_by_S`, оно может факторизоваться через core: verifier действительно устанавливает собственный статус принятия. Если целевое поле — отдельно заданная запись о предназначенном применении, пригодности физического сопряжения либо цели, конфликтующее волокно исключает автономный выбор этого поля.

12. Применение и цель как независимые координаты

В одном пакете могут быть заданы разные координаты:

$$Q_A : K \rightarrow Y_A, Q_V : K \rightarrow Y_V.$$

Например, две допустимые записи могут иметь одинаковый математический core и различаться строками поля Q_A :

$$\ulcorner \text{«ракета летит»} \urcorner, \ulcorner \text{«ракета не летит»} \urcorner.$$

Эти значения являются строками протокола. Теорема не выбирает одну из них и не удостоверяет названное.

Даже если Q_A зафиксировано одинаково, поле Q_V может содержать различающиеся строки:

Γ «цель принята» $\neg$, Γ «цель не принята» $\neg$.

При одинаковом доступном входе конфликтующее волокно даёт отдельное $\neg \text{AutExport}(K, \rho, Q_v)$.

Положительная и отрицательная строки симметричны. Отсутствие goal-input не устанавливает отсутствие смысла; оно устанавливает лишь невозможность получить заданную целевую координату из предъявленного core при наличии конфликтующей пары.

13. Точные фальсификаторы

Фальсификатор пары

Конкретный свидетель рушится, если:

$$\rho(k_0) \neq \rho(k_1)$$

или

$$Q(k_0) = Q(k_1).$$

Фальсификатор локальности процедуры

Если заявлено $T = T' \circ \rho$, но при одинаковом ρ наблюдаются разные точные трассы, то процедура прочитала необъявленный вход, использовала незафиксированное состояние либо была неверно описана.

Фальсификатор факторизации канала

Если заявлено $\eta = h \circ \rho$, но η различается на одном волокне ρ , заявленная факторизация ложна.

Случаи, где определение возможно

No-selector заключение не применяется, если:

- Q постоянно на каждом волокне;
- достаточная для Q запись уже входит в ρ ;
- добавлен канал, разбивающий все конфликтующие волокна;
- заранее сужен класс K до однородных волокон;
- интерактивный ответ вносит различающий вход;
- seed или выбор ветви несёт зависимость от индекса.

Каждый такой случай изменяет проекцию, допустимый класс или источник различия. Он не является ещё одной функцией прежнего core.

14. Граница притязаний

RBI-1 полностью устанавливает:

Если два независимо заданных индекса имеют один и тот же полный записанный вход и разные целевые записи, то никакая функция этого входа — независимо от сложности, длины, внутренней семантики и формального аппарата — не восстанавливает целевую координату на обоих индексах. Успешное разделение требует добавленного различающего канала либо добавленного ограничения класса.

RBI-1 не устанавливает:

- какая целевая запись истинна;
- существует ли что-либо, названное целевой записью;
- что целевая строка относится к чему-либо помимо строки;
- что формальный вывод недействителен;
- что применение, цель или смысл отсутствуют;
- что всякая целевая координата имеет конфликтующее волокно;
- историческое происхождение математики.

Глобальный вывод для класса координат Q требует отдельного свидетеля для каждой координаты:

$$\forall Q \in Q \exists k_0, k_1 : \rho(k_0) = \rho(k_1) \wedge Q(k_0) \neq Q(k_1).$$

Только при этом условии:

$$\forall Q \in Q : \neg \text{AutExport}(K, \rho, Q).$$

15. Максимальная точная строка

Вычисление может сколько угодно преобразовывать доступную запись. Оно не может разделить два индекса, которые этой записью не разделены. Любое успешное разделение предъявляет новый канал или новое ограничение.

Для публичного названия «Математика не вышла из записи» выражение «не вышла» означает только формулу RBI-NS относительно объявленных K, ρ, Q . Само название без этой области получает статус UNBOUNDED-STANDALONE.

AB-1: применение, пригодность и цель как добавленные записи

1. Назначение

RBI-1 отвечает на общий вопрос: может ли вычисление разделить целевую координату, если полный доступный core её не разделяет. AB-1 задаёт более узкий операционный аудит математического применения.

AB-1 не удостоверяет связь записи с названным. Он проверяет только, была ли до границы τ предъявлена полная запись, где заявленное применение снабжено точными зависимостями.

2. Класс конечных проверяемых математических записей

Элемент класса FVMR имеет вид

$$X = \langle W, I, F, \pi, \varphi, V, \tau \rangle,$$

где:

- W — точный конечный carrier;
- I — контракт чтения;
- F — внутренний формальный контракт;
- π — конечный proof-object;
- φ — записанное заключение;
- V — verifier, завершающийся на данном входе;
- τ — последний включённый индекс трассы.

Условие членства:

$$V(W, I, F, \pi, \varphi) \downarrow \in \{0, 1\}.$$

Конечность относится к спецификации и каждому проверяемому объекту. Множество выражений, принимаемых системой, может быть бесконечным. Если точный carrier, контракт или завершающийся verifier не предъявлены, статус — OUTSIDE-FVMR, а не отрицательный вердикт о неназванном объекте.

3. Необратимый порядок слоёв

В трассе различаются:

Слой	Максимальный протокольный статус
W	CARRIER-SUPPLIED
$W + I$	READ-AS-MATHEMATICAL
$W + I + F + \pi$	FORMALLY-CHECKED
$+ B$	BRIDGE-CLAIM-REGISTERED
$+ A$	APPLICATION-BINDING-RECORDED
$+ O$	FIT-ASSESSMENT-RECORDED
$+ G$	PURPOSE-RECORD-ADDED

Ни один поздний статус не переносится назад в ранний carrier. Слово bridge означает зарегистрированное притязание на связь, а не саму связь.

4. Полная application-binding запись

Binding для математического результата m задаётся семью полями:

$$a = \{ m, b, t, \sigma, u, s, c \},$$

где:

1. m — точный идентификатор математического результата;
2. b — идентификатор ранее зарегистрированной bridge-записи;
3. t — идентификатор либо span целевой записи;
4. σ — записанное отображение формальных символов на поля целевой записи;
5. u — записанное предполагаемое использование;
6. s — записанные область, условия и горизонт;
7. c — записанный критерий пригодности.

Условие Complete(a) означает только:

- все семь полей присутствуют и непусты;
- ссылки разрешаются в пределах объявленной трассы;
- m указывает на ранее принятый formal-result;

- b указывает на ранее зарегистрированный bridge-claim;
- t указывает на точный carrier/span;
- σ имеет объявленную структуру;
- зависимость от b не удалена.

Complete(a) не означает, что целевая запись истинна, что bridge корректен, что названное существует или что применение пригодно.

5. Тотальный статус до границы

Для валидной конечной трассы:

$$AppStatus_{C,\tau}(m) \in \{APPLICATION - SPECIFIED, APPLICABILITY - UNSPECIFIED\}.$$

Статус APPLICATION-SPECIFIED присваивается тогда и только тогда, когда до τ существует принятый полный binding для m . Во всех остальных валидных случаях:

$$AppStatus_{C,\tau}(m) = APPLICABILITY - UNSPECIFIED.$$

Диагностические коды:

- NO-BINDING;
- MISSING-BRIDGE;
- UNKNOWN-MATH-ID;
- MISSING-TARGET-RECORD;
- MISSING-SYMBOL-MAP;
- MISSING-USE;
- MISSING-SCOPE;
- MISSING-CRITERION;
- REFERENCE-AFTER-TAU.

APPLICABILITY-UNSPECIFIED означает ровно отсутствие полной допустимой binding-записи в предъявленной трассе до τ . Он не означает NOT-APPLICABLE, ложность, отсутствие названного или невозможность будущего применения.

6. Ненаследование применимости

Пусть продолжение $\tau' \geq \tau$ по контракту AB-1 не добавляет узлов B/A и не регистрирует зависимости семи binding-полей. Тогда по определению AppStatus:

$$AppStatus_{C,\tau}(m) = APPLICABILITY - UNSPECIFIED \Rightarrow AppStatus_{C,\tau'}(m) = APPLICABILITY - UNSPECIFIED$$

Это протокольное ненаследование основано на точной read-set и структуре AB-1. RBI-1 отдельно устанавливает: если два допустимых расширения одной математической проекции различаются target/binding-полями, никакая функция этой проекции не выбирает требуемое расширение на обоих индексах. RBI-1 не запрещает напечатать один и тот же candidate binding на всех индексах; такой payload получает application-status только после регистрации требуемых узлов и зависимостей.

7. Парные аудиты

7.1. Binding отсутствует / binding добавлен

Две трассы могут иметь одинаковую математическую проекцию:

$$\pi_M(\tau_0) = \pi_M(\tau_1),$$

но в τ_1 после математической части добавлены bridge и полный binding. Статусы различаются. Следовательно, статус применения не является функцией одной математической проекции.

7.2. Два target-record

При одной математической проекции два допустимых расширения могут содержать разные значения t . Если контракт допускает оба, RBI-1 запрещает математической трассе выбрать одно из них автономно.

7.3. Два использования

При одинаковых m, t, σ могут различаться строки поля u , например predict и control. Сам формальный результат не выбирает использование.

7.4. Payload spoof

Математический payload может побайтно содержать слова bridge, application, ракета, летит, цель ИЛИ полную сериализацию binding-записи. Это не создаёт зарегистрированную зависимость и не заполняет AB-1. Содержание payload не получает право назначить себе статус.

8. Максимальная уступка о ракете

Допускаются как последующие строки:

«расчёт выполнен»;
 «ракета построена»;
 «ракета существует»;
 «ракета полетела»;
 «траектория совпала с расчётной».

Уступка принимает только наличие этих строк в расширенном входе. Она не переносит ни одну из них внутрь более раннего proof-object.

Различаются:

m — запись расчёта; b — запись bridge-claim; a — application-binding запись; o — запись, классифицированная как телеметрия; $f = \ulcorner \text{«ракета полетела»} \urcorner$; g — goal-record.

Не допускается сокращение $m = a = o = f = g$.

Даже если строка f принимается без спора, она не указывает автоматически:

- какой formal-result к ней привязан;
- какое отображение связывает символы с полями записи;
- какая процедура классифицирует вход как телеметрию;
- в какой области заявлена пригодность;
- какой критерий совпадения применён;
- какое использование и какая цель выбраны.

Все эти различия добавляются отдельными полями или каналами.

Короткая формула:

Расчёт мог быть записан раньше полёта.

Полёт не был строкой расчёта.

Это не отрицание причинного действия. Цепочка

record $\rightarrow$ interpreter $\rightarrow$ controller $\rightarrow$ actuator

может быть внесена в расширенный протокол. Тогда interpreter, controller, actuator, их входы и trust-условия образуют добавленный канал и implementation-зависимость. Они не появляются как ещё один вывод прежней формулы.

9. «Зачем?»

Цель отделяется от формальной выводимости, применения и пригодности:

FORMALLY - CHECKED

- ⇒ APPLICATION - SPECIFIED
- ⇒ FIT - ASSESSMENT - RECORDED
- ⇒ PURPOSE - RECORD - ADDED
- ⇒ PURPOSE - JUSTIFIED

Каждый знак $\neq$ означает отсутствие наследования статуса внутри объявленного контракта. Он не означает невозможность позднего добавления.

Записи 100, 200, 300 лет могут стать параметрами только по объявленному контракту чтения. Увеличение числовой строки не добавляет goal-record:

$$M + \ulcorner 100 / 200 / 300 \urcorner \not\vdash G.$$

Если функция полезности или цель включена в формальный core, математика может вычислять следствия относительно этой записи. Она переносит заданный критерий через правила; она не удостоверяет его желательность и не делает его единственной целью без отдельного условия допустимости.

Точная симметричная строка:

Расчёт не установил ни смысл, ни отсутствие смысла. Обе противоположные строки требуют отдельного целевого поля; при конфликтующем волокне ни одна не выбирается вычислением core.

10. Model theory и proof assistants

Формальная система может корректно устанавливать:

$$\Gamma \vdash \varphi, S \models \varphi,$$

либо свойства программы относительно записанного I/O-контракта. AB-1 не отрицает эти условные результаты.

Разделяются четыре слоя:

1. синтаксическая выводимость относительно Γ ;
2. удовлетворимость в точно заданной структуре S ;
3. адекватность S предназначенному применению;
4. принятие цели или критерия.

Первые два могут полностью входить в formal-core. Третий требует application/fit input. Четвёртый требует goal/endorsement input. Если все четыре записаны в одной большой формальной системе, условные следствия вычисляются внутри неё; но статусы intendedness, physical coupling и endorsement не следуют в этом контракте из одного присутствия соответствующих строк.

11. Граница τ

Фиксация результата:

$$\tau^* = \langle h_C, h_V, h_{trace}, n, d \rangle,$$

где n — последний включённый индекс, а d — строка времени фиксации.

Если до τ^* полного binding нет, записывается:

APPLICABILITY-UNSPECIFIED-AT-TAU

Позднее добавление полного binding создаёт новый статус на $\tau' > \tau^*$:

APPLICATION-SPECIFIED-AT-TAU-PRIME

Старый результат не переписывается: он относился к более раннему входу.

12. Граница притязаний

AB-1 устанавливает наличие, отсутствие и происхождение application-binding записей. RBI-1 устанавливает невозможность автономно выбрать конфликтующую целевую координату из одинакового core.

Ни один из результатов не устанавливает:

- наличие или отсутствие названного;
- истинность bridge-claim;
- физическую причинность;
- фактический успех применения;
- цель, смысл или их отсутствие;
- историческое происхождение математики.

Максимальная строка:

Формальный proof-object закончился статусом внутри своего контракта. Применение, пригодность и цель начались только там, где в доступный вход были добавлены различающие записи.

Самоприменение Phase 1

1. Никакого исключения для отчёта

Рукопись, формулы, код, тесты и публичное название Phase 1 являются carrier-вхождениями. Их чтение как исследования, теоремы, проверки или вывода производится по объявленным контрактам. Они не предъявляют названное собственной силой.

Порядок собственного пакета:

$$W_p < I_p < F_p < R_p < A_p.$$

Здесь:

- W_p — точные байты файлов;
- I_p — правила декодирования, schema и канонизации;
- F_p — RBI-1, lift-контракт и исполняемые checkers;
- R_p — записанные статусы запусков;
- A_p — любые дальнейшие строки о применении результата.

Ни один поздний слой не переносится назад.

2. Собственный конфликтующий пример

Пусть точный текст максимальной строки Phase 1 является общим core:

Никакое вычисление одинакового доступного core не выбирает различающуюся целевую координату.

Два допустимых extension-record могут присвоить полю report_applies_to_unbounded_mathematics разные булевы значения при одинаковых байтах рукописи. Если оба значения разрешены self-contract, RBI-1 запрещает рукописи автономно выбрать одно из них.

Это не опровержение теоремы. Теорема имеет точную область (K, p, Q) ; unbounded extension-claim не входит в её вывод.

3. Разрешённые собственные статусы

Ключевые строки получают не более следующих статусов:

Строка/артефакт	Допустимый статус
доказательство RBI-F/RBI-NS	TEXTUAL-THEOREM-IN-DECLARED-METATHEORY
принятый JSON-свидетель	VERIFIED-IN-DECLARED-ARTIFACT
результат unit tests	TEST-RUN-RECORDED
точный SHA-256	BYTE-IDENTITY-RECORDED
Phase 0 baseline	BASELINE-MANIFEST-REVERIFIED
публичное название	UNBOUNDED-STANDALONE
сравнительное картирование без T0	INDETERMINATE-NO-PRIORITY-CLAIM

Запрещённые автоматические статусы:

REFERENT - PRESENTED
WORLD - ATTESTED
TRUE - INDEPENDENTLY - OF - METATHEORY
SELF - CERTIFIED
COMPLETE - FOR - ALL - MATHEMATICS
HISTORICAL - PRIORITY - ESTABLISHED
PURPOSE - JUSTIFIED

4. Общая теорема и конечная проверка

Python checker может исчерпывающе проверить конечный перечисленный K : сгруппировать элементы по canonical core и проверить постоянство target внутри каждого волокна. Он не квантифицирует машинно по всем множествам, функциям и селекторам.

Поэтому различаются:

TEXTUAL - THEOREM - IN - DECLARED - METATHEORY

и

VERIFIED - IN - DECLARED - ARTIFACT

Первый относится к ручному общему доказательству. Второй — к точному конечному входу и зафиксированному checker.

5. Trusted computing base

Воспроизводимый статус зависит от:

- точных байтов schema, checker, fixture и test suite;
- Python runtime, названного в TEST_REPORT;
- JSON parser и Draft 2020-12 validator;

- файловой системы и запускающего окружения;
- корректности генерации и чтения manifest;
- визуального и структурного контроля DOCX/PDF;
- ручного proof review общей теоремы.

Хеш подтверждает тождество байтов, а не истинность содержания. Unit test подтверждает результат конкретного исполнения, а не отсутствие всех возможных дефектов.

6. Самоприменение к слову «математика»

Строка `математика` в названии не задаёт универсальный домен. В научном результате охватываются:

1. произвольные функции объявленной доступной проекции — в RBI-1;
2. конечные verifier-системы — в lift-контракте;
3. опубликованные fixtures — в машинных проверках;
4. application-binding записи — в AB-1.

Неформальные, скрытые, незавершающиеся или незафиксированные практики не объявляются ложными. Для их включения требуется точная проекция и target-coordinate. Публичная жёсткость названия не расширяет формальный квантор.

7. Самоприменение к слову «доказательство»

Форма «теорема доказана» внутри рукописи получает разбор:

- exact span строки;
- ссылка на объявленные определения;
- конечная последовательность переходов;
- область кванторов;
- список TCB;
- статус TEXTUAL-THEOREM-IN-DECLARED-METATHEORY.

Она не получает самостоятельного перехода к external truth. Это не делает доказательство бесполезным: оно точно фиксирует следование формулы из объявленных посылок. Оно лишь не освобождается от собственной границы.

8. Самоприменение к «ракете» и «зачем?»

Все употребления форм ракета, летит, Марс, 100/200/300 лет, цель, смысл в пакете являются либо значениями fixture, либо цитируемыми строками, именами целевых полей или максимальными уступками в аргументе.

Ни одно употребление не получает статус event record без отдельного контракта. Отрицательные формы имеют ту же границу.

9. Хронология и новизна

Phase 0 зафиксирована до сравнительного картирования и не изменяется. Phase 1 создана как отдельное post-freeze продолжение. Независимость KO от сравнительных источников сохраняется; ни одна comparison-only запись не входит в доказательство RBI-1. Phase 1 не утверждает, что её новые формулировки завершены до чтения всех кандидатов. Отсутствие внешнего публичного TO запрещает притязание на исторический приоритет; novelty status остаётся INDETERMINATE до исполнения PFNP-1.

Это различие не скрывается общим словом «исследование».

10. Точная собственная граница

Максимальный собственный вывод:

*В объявленной метатеории доказана факторизационная и no-selector теорема.
Опубликованные конечные fixtures приняты зафиксированными checkers. Эти статусы не
предъявляют названное строками отчёта и не расширяют область за объявленные
K, p, Q, код и TCB.*

Публичная строка:

Математика не вышла из записи.

Без подзаголовка и формулы RBI-NS получает только:

UNBOUNDED - STANDALONE

Научная строка:

*При конфликтующем волокне никакая функция полного доступного core не
восстанавливает целевую координату на всех допустимых индексах.*

Получает:

TEXTUAL - THEOREM - IN - DECLARED - METATHEORY

Враждебная проверка Phase 1

1. Условие прохождения

Phase 1 проходит проверку только в том случае, если центральный результат не зависит от переименования записей, заранее запрещённого правила $m \rightarrow A$, авторитета verifier, истинности bridge или от отрицания семантики.

Основание RBI-1 должно сводиться к четырём независимо проверяемым пунктам:

1. объявлено множество индексов K ;
2. объявлена доступная проекция ρ ;
3. целевая координата Q задана независимо;
4. предъявлена пара с равным ρ и различным Q .

После этого no-selector выводится для любой функции доступного входа.

2. Снятые блокирующие возражения

2.1. «Знак не тождествен референту» не значит «знак ни к чему не относится»

RBI-1 не выводит отсутствие отношения из нетождественности. Он проверяет более узкое утверждение: если целевая координата различается внутри одного волокна доступного входа, то она не является функцией этого входа. Отношение может быть добавлено каналом η ; его корректность является отдельным притязанием.

2.2. Теорема не должна быть истинной по типовой политике

MRB-типизация остаётся реализационным контролем. Универсальный результат доказывается до неё через

$$\rho(k_0) = \rho(k_1), Q(k_0) \neq Q(k_1).$$

Даже система с внутренними строками A, V, True, Denotes и Applies подпадает под RBI-1, если все эти строки вычислены из того же core.

2.3. Bridge не выдаётся за grounding

Bridge — запись о заявленной связи и узел provenance. Проверка его наличия не устанавливает связь с названным, истинность целевой строки или надёжность trust anchor.

2.4. Model theory не отрицается

Результаты $\Gamma \vdash \varphi$ и $S \models \varphi$ полностью допускаются относительно записанных Γ, S и правил. Целевой вопрос intendedness структуры, физической адекватности или принятия цели задаётся отдельным Q . Только для конфликтующего волокна такого Q применяется no-selector.

2.5. Причинная эффективность не отрицается

Запись может входить в цепь управления. Тогда interpreter, механизм исполнения, входы среды и сопряжение являются различающими каналами. Тезис Phase 1 — не «запись не действует», а «действие и его пригодность не являются автономным выводом замкнутого proof-core».

2.6. Исторический тезис не является посылкой

Фраза «математика возникла из языка» не используется как исторический факт и не служит генетическим опровержением. Проверяемый порядок локален:

carrier < reading contract < formal trace < application binding < goal record.

3. Критические различия

Следующие позиции нельзя смешивать:

Позиция	Что фиксируется	Чего статус не добавляет
physical token/carrier	точные байты, пиксели, спаны или иной carrier	значение, proposition, референт
parsed expression	результат объявленного parser/reading contract	истинность выражения
proof acceptance	соответствие proof-object правилам verifier	intended-world adequacy
model satisfaction	результат относительно заданной структуры	что структура является предназначенной
application binding	записанное отображение и область	истинность связи
implementation channel	записанная цепь исполнения	желательность эффекта
goal/utility	заданный критерий	оправданность критерия

Конечный token может кодировать выражение о бесконечном объекте. Phase 1 не выводит ограниченность содержания из размера носителя. Конечность нужна только для воспроизводимого аудит-пакета и завершающегося verifier на данном входе.

4. Точные фальсификаторы RBI-1

4.1. Пара не является парой одного волокна

Если канонические core различаются, свидетель не доказывает no-selector.

4.2. Target не различается

Если целевые значения равны, данная пара является determined-control, а не конфликтующим волокном.

4.3. Процедура прочитала больше, чем заявлено

Если трассы различаются на одинаковом core, либо нарушена детерминированность, либо read-set неполон. Процедура должна быть переклассифицирована с расширенной проекцией.

4.4. Целевая координата изменена после результата

Если определение Q , допустимость элементов или проекция изменены после просмотра вердикта без новой версии контракта, аудит недействителен.

4.5. Достаточный канал уже был во входе

Если поле, различающее все target-конфликты, уже входило в ρ , заявленная конфликтующая пара не сохраняет одинаковый core.

5. Выходы из заявленного класса

Не входят в исполняемый FVMR-аудит без дополнительной фиксации:

- бесконечный proof-object, не представленный конечным сертификатом;
- незавершающаяся проверка данного входа;
- скрытое устройство, состояние, сеть или oracle;
- случайность без записанного seed/распределения;
- интерактивность без полного transcript/read-set;
- социальная проверка без точного журнала и решения;
- неформальное доказательство без объявленного verifier-контракта.

Если дополнительные входы фиксируются, они входят в расширенную ρ , и RBI-1 применяется заново.

6. Возражения и точные ответы

«Формула сразу написана про ракету»

Есть `parsed expression` с соответствующим `token`. Чтобы получить `application claim`, требуется показать, что целевая координата факторизуется через доступный `input` либо добавить различающий канал. Само написание `token` не выполняет этот шаг.

«Единицы измерения уже указаны»

`kg`, `m`, `s` могут быть символами внутреннего контракта. `Measurement procedure`, `calibration`, `sample`, `timestamp` и `mapping` являются отдельными полями, если они нужны целевому притязанию.

«Эксперимент подтвердил модель»

Тогда вход расширен `observation-record`, процедурой и критерием. Этот канал может разделить прежнее волокно. Результат принадлежит расширенному контракту, а не одной математической деривации.

«Proof assistant гарантирует программу»

Он гарантирует формализованное следствие относительно `kernel` и `I/O-contract`. Аппаратное сопряжение, свойства среды и желательность спецификации либо входят в `trusted input`, либо остаются отдельными координатами.

«Вся семантика уже формализована»

Формализованная семантика является частью $T(\rho)$. Если `intendedness` различается на одном волокне ρ , внутреннее вычисление совпадает и не выбирает `target`. Если `intendedness` внесена в `core`, проекция изменилась.

«Математик понимает применение без записи»

Непредъявленный `input` не входит в воспроизводимый `read-set`. Его предъявление расширяет ρ . Phase 1 не отрицает понимание; она запрещает считать скрытый вход результатом публичного `proof-object`.

«Вопрос “зачем?” не относится к математике»

Именно это различие и фиксируется. Математика может оптимизировать заданную `utility-функцию`. Она не выбирает и не оправдывает `utility-функцию` из одного расчёта, если допустимый класс содержит конфликтующее `goal-волокно`.

7. Самопроверка и TCB

Собственная теорема Phase 1 существует как текстовое доказательство в объявленной метатеории. Python-checkers удостоверяют только конечные JSON-контракты и fixtures. Они зависят от:

- точных байтов исходников;
- Python runtime;
- JSON parser и schema validator;
- операционной среды;
- корректности ручного доказательства общего случая.

Ни тесты, ни хеши не дают универсальной self-certification. Общая теорема получает статус TEXTUAL-THEOREM-IN-DECLARED-METATHEORY; конечные fixtures — VERIFIED-IN-DECLARED-ARTIFACT.

8. Языковой стоп-лист

Рукопись не замораживается, если в ней без явной протокольной области появляются формы:

- «формула ни к чему не относится»;
- «референции не существует»;
- «математика не действует»;
- «bridge доказывает связь»;
- «измерение предъявило событие»;
- «смысла нет»;
- «истина невозможна»;
- «вся математика полностью опровергнута»;
- «впервые» без завершённого post-freeze поиска и внешнего T0.

Допустимая сильная форма:

Никакое вычисление одинакового доступного core не выбирает различающуюся целевую координату. Успешное различение всегда показывает добавленный канал либо суженный класс допустимости.

9. Финальный adversarial verdict

При точных условиях RBI-1 контрпримером не является более мощная математика. Контрпример должен сделать одно из трёх:

1. разделить одинаковый вход детерминированной функцией только этого входа;
2. показать, что опубликованная пара имела разные core или одинаковые target;
3. показать, что целевая координата не была независимой от вердикта.

Первый вариант противоречит функциональности. Второй и третий являются точными фальсификаторами конкретного аудита. Поэтому результат закрыт в объявленной области и остаётся открытым только там, где область, read-set или target были изменены.

PFNP-1: предварительный collision-аудит

1. Роль сравнения

Сравнительный проход выполнен после фиксации Phase 0. Все внешние записи имеют только:

```
role: comparison_only
admissible_as_premise: false
```

Ни одна запись R01–R13 не является ссылкой RBI-1, AB-1, lift-контракта или self-audit. Сравнение может изменить только status новизны уже собранной конфигурации.

2. Проверенный предварительный корпус

В полном тексте с точными локаторами проверены тринадцать записей:

- R01 Hilbert (1926), 3; R02 Wittgenstein (1922), 2; R03 Tarski (1944), 3; R04 Hempel (1945), 4; R05 Harnad (1990), 4;
- R06 LF (1993), 1; R07 Rushby (1992), 1; R08 Appel (2001), 3; R09 provenance semirings (2007), 1;
- R10 core provenance (2013/2014 copy), 2; R11 Coq 8.9.0 (2019), 1; R12 MetaCoq (2019/2020), 2; R13 Lean 4 (2021), 2 при $F_3=U$ и maximum score 3.

Наибольшее подтверждённое частичное столкновение равно четырём признакам у R04 и R05. Во всех R01–R13 $F_4=F_8=F_9=F_{10}=0$. Для R13 единственная неразрешённая ячейка полного mapped record — $F_3=U$.

Это не доказывает отсутствие полного совпадения в литературе. Полная матрица F_1 – F_{10} , точные spans, URL, SHA-256 проверенных копий и причины каждого 0/U сохранены в отдельном воспроизводимом файле 05_COLLISION_AUDIT_RU.md.

3. Самые близкие частичные линии

Hempel и Harnad разделяют carrier/sign и интерпретацию, требуют отдельный bridge и не переносят позднюю интерпретацию в исходный token. В проверенных записях отсутствуют совместно typed proof provenance, dual autonomy test, extension witness, self-audit и reproducibility package.

Appel связывает machine word, decoding, typed proof и safety policy. Но bridge/goal identities не сохраняются как отдельные provenance-узлы и нет отдельных операций A-INTRO/V-INTRO, требуемых замороженным PFNP-1.

Rushby даёт близкий noninterference subfeature; Green, Acar, Coq, MetaCoq и Lean закрывают отдельные механизмы proof/provenance. Ни одна из этих частичных линий не засчитана как вся конфигурация.

4. Неразрешённый остаток

Остаются U01–U08: Goguen–Meseguer, dependency calculus Abadi et al., Putnam, Benacerraf, Skolem, поздний Wittgenstein, длинная версия Tarski и Nacula PCC. Не выполнены обязательные query families для каждого F1–F10, систематический book/dissertation/repository pass, две saturation iterations и разрешение всех U.

Backward/forward citation pass не запускался по правилу PFNP-1: ни один подтверждённый кандидат не достиг пяти единиц.

5. Честный итог

Внешний публичный (T₀) для Phase 0 не создан. Поэтому локальная дата freeze не сравнивается как доказательство исторического приоритета. Правило остановки PFNP-1 не выполнено.

```
no_full_configuration_observed_in_R01_R13: true
stopping_rule_satisfied: false
external_public_T0_anchor: null
priority_claim_authorized: false
novelty_status: INDETERMINATE
```

Допустимая формулировка:

В предварительно проверенных R01–R13 не установлена более ранняя запись, содержащая всю конфигурацию F1–F10 и J. Поскольку внешний T0 отсутствует, правило остановки не выполнено и остаток U открыт, статус новизны остаётся INDETERMINATE.

Недопустимы: «аналогов нет», «вся предшествующая литература проверена», «приоритет доказан» и NO_FULL_CONFIGURATION_IN_MAPPED_CORPUS.