

---

# ОТСУТСТВИЕ ЗАПИСИ НЕ ЯВЛЯЕТСЯ ЗАПИСЬЮ ОТСУТСТВИЯ

Теорема максимальных продолжений, отрицательное происхождение и  
граница доказательства отсутствия

Кирилл Шебетов · beforeword / research

Phase 4 · technical version 4.0.0 · 29 августа 2026

## Содержание

### Аннотация

- 0. Нулевая уступка отрицательным словам
  - 1. Точный объект Phase 4
  - 2. Линия Phase 0 → Phase 1 → Phase 2 → Phase 3 → Phase 4
  - 3. Finite positive support system
  - 4. Extension alphabet и объявленная семья
  - 5. Две леммы, на которых держится CBE-4
  - 6. CBE-4: теорема максимальных продолжений
  - 7. Положительный control: конечное отсутствие действительно доказывается
  - 8. MCW-4: минимальные контрпродолжения
  - 9. Минимальный разрушающий пример
  - 10. Отрицательный code не является nonmembership
  - 11. Closed-world переход как новый конструктор
  - 12. BND-4: граница продолжений является input
  - 13. Исчерпывающая семья и предел слова exhaustive
  - 14. Доказательство от противного и closed-world policy
  - 15. Ракета, зачем и 100/200/300 лет
  - 16. Сильнейшие возражения
  - 17. Admission gates и обязательные controls
  - 18. Точные фальсификаторы
  - 19. Самоприменение
  - 20. Claim/status и источники
  - 21. Граница притязаний
  - 22. Итог
- Статус источников

### 01. ТЕОРЕМА ПРОИСХОЖДЕНИЯ ОТРИЦАТЕЛЬНОГО СТАТУСА

#### НЕ НАЙДЕНО НЕ УСТАНОВЛИВАЕТ ОТСУТСТВИЕ

- 0. Нормативная область и нулевая уступка словам
- 1. Конечный типизированный completion-boundary contract
- 2. Положительное типизированное замыкание
- 3. Три уровня отрицательного результата
- 4. Полярностное разделение
- 5. CBE-4: теорема эквивалентности границы завершения
- 6. MCW-4: теорема минимального контррасширения
- 7. Completion boundary является различающим input
- 8. Admission gates отрицательного status
- 9. Положительные controls
- 10. Точные falsifiers и условия снятия

11. Абстрактное расширение и его обязательные условия
12. Самоприменение и граница результата

#### **CWC-4: сертификат полноты конечного класса дополнений**

0. Нормативный статус
1. Три несовпадающих статуса
2. Конечный completion-contract
3. Монотонность дополнения
4. Максимальный фронтير семейства
5. Минимальные контрдополнения
6. Каноническое связывание с input
7. Детерминированные closure-свидетельства
8. CWC-4: теорема точного сертификата
9. Exact schema/runtime mapping
10. Опциональная hitting/cut-двойственность
11. Контрольные конструкции
12. Adversarial falsifiers сертификата
13. Self-boundary
14. Итоговый нож Phase 4

#### **BND-4: ГРАНИЦА ПРОДОЛЖЕНИЙ, DEFAULT И ОТМЕНА**

0. Нормативный статус
1. Положительное ядро Phase 4
2. Три разных отрицательных операции
3. BND-4: граница продолжений является входом
4. REV-4: default verdict относится к snapshot
5. DEF-4: порядок defeat является входом
6. Почему maximal-frontier theorem сюда не переносится автоматически
7. Отрицательный code не является дополнением автоматически
8. Proof by contradiction и default negation
9. Exception и изменение правил
10. Backdating, timestamp и история
11. Closed-world как exact special case
12. Ракета, цель и justification
13. Сто, двести, триста snapshot
14. Самоприменение BND-4
15. Admission gates
16. Точные фальсификаторы
17. Максимальная формулировка

#### **ПРОТИВОРЕЧИЕ НЕ ЯВЛЯЕТСЯ ОТСУТСТВИЕМ**

Формальное отрицание, исчерпывающий поиск, ракета и граница NOT-FOUND  
Аннотация

0. Сначала предъявлены строки
1. Точный объект применения
2. Exact apparatus CBE-4
3. Семь различных отрицательных поверхностей
4. Formal negation полностью допустимо
5. Proof by contradiction не является ошибкой
6. Complement требует exact universe
7. Исчерпывающий конечный positive control
8. Local NOT-FOUND и минимальный counterextension
9. Timeout, failure и negative response
10. Ракета: максимальная техническая уступка
11. Goal, endorsement и SENI-4 как application corollary
12. Justification и отсутствие возражения
13. Сто, двести и триста лет
14. Максимальные возражения
15. Self-application
16. Итоговый разрез

#### **05. ВРАЖДЕБНАЯ ПРОВЕРКА И САМОПРИМЕНЕНИЕ PHASE 4**

##### **ОТСУТСТВИЕ ЗАПИСИ НЕ ЯВЛЯЕТСЯ ЗАПИСЬЮ ОТСУТСТВИЯ**

0. Назначение и нулевая уступка словам
1. Exact object и admission gates
2. CBE-4, MCW-4 и BND-4
3. Missing maxima и cofinality
4. Немонотонный target, default и defeat
5. Incomplete search, omitted extension и timeout
6. Post hoc boundary
7. Empty, malformed и inconsistent family
8. MinCE и support-basis нельзя склеивать
9. Negative-code conflation
10. Обязательные positive controls
11. Mutable history и payload spoof
12. Rocket/export
13. Самоприменение
14. Exact theorem falsifiers
15. Implementation falsifiers
16. Claim registry и границы
17. Финальная граница красной команды

## Аннотация

Для finite positive support system с конечным носителем claim-codes  $N$ , конечным носителем protected base-ids  $B = S \dot{\cup} A$ , current bases  $S$ , candidate bases  $A$ , и positive rules  $R$  объявляется конечная непустая семья

$$\emptyset \in E \subseteq P(A).$$

Каждое  $E \in E$  добавляет base-ids и задаёт closure claim-codes  $C_E = Cl_R(S \cup E)$ . **CBE-4** доказывает: code  $q$  отсутствует во всех  $C_E$  тогда и только тогда, когда он отсутствует в closure каждого  $M \in Max(E)$ , family counterextensions пуста и её минимальный antichain пуст. Это положительный результат: точная finite family позволяет полностью доказать code-absence в собственном домене. **MCW-4** характеризует ложность этого статуса через минимальные по включению counterextensions.

Непринадлежность  $q^+ \notin C_E$  не является принадлежностью отдельного  $q^- \in C_E$ . Negative-code требует собственного base-activated introduction либо rule с конечным support tree; closed-world переход является новой policy. **BND-4** показывает: при одном core разные  $E$  могут требовать разных absence-status, поэтому boundary входит в input. Доказанный ABSENT-IN-DECLARED-COMPLETION-CLASS не экспортируется в named/world absence без нового binding и scope-record.

**Ключевые слова:** граница записи; отсутствие; отрицательный code; максимальное продолжение; минимальное контрпродолжение; finite closure; completeness boundary; closed-world policy; provenance; самоприменение; цель.

## 0. Нулевая уступка отрицательным словам

Сначала предъявлены строки:

```
нет
отсутствует
не существует
невозможно
не доказано
не найдено
бессмысленно
цели нет
```

Ни одна из них не предъявляет отсутствие, несуществование, невозможность, недоказанность или отсутствие цели. Отрицательная грамматическая форма остаётся положительно присутствующей последовательностью знаков. Частица не не выносит строку из record и не превращает её в доступ к названному.

Та же остановка применяется к техническим словам этой рукописи. Множество, closure, maximal, extension, absence, exhaustive, proof и world являются метками объявленного reading contract. Слово exhaustive не исчерпывает ничего одним написанием. Слово world не предъявляет мир. Запись  $q \notin X$  получает смысл только относительно точного carrier  $N$ , точного множества  $X$  и объявленной операции сравнения.

Поэтому исследование не начинается с фразы «объекта нет». Оно начинается с конечных code, конечных правил и перечисленной семьи extensions. Сильнейший отрицательный вывод сначала формулируется так:

*code  $q$  не входит ни в одно closure, построенное из членов точной объявленной семьи  $E$ .*

Это полноценный универсальный вывод по конечному объявленному домену. Его нельзя ослаблять до «ничего не доказано». Его также нельзя расширять до отсутствия названного, не добавив новый переход.

## 1. Точный объект Phase 4

Phase 4 различает четыре записи, которые обычно склеиваются одним отрицательным словом:

1.  $q$  не появился в текущем closure при  $E = \emptyset$ ;
2.  $q$  не появился в closure отдельно выбранного extension;
3.  $q$  не появился ни в одном closure объявленной семьи;
4. отдельный negative-code  $q^-$  появился в closure.

Между ними нет автоматических тождеств. В частности,

$$q \notin Cl_R(S) \not\Rightarrow q \notin Cl_R(S \cup E),$$

поскольку правило может вывести  $q$ , и

$$q^+ \notin Cl_R(S \cup E) \not\Rightarrow q^- \in Cl_R(S \cup E),$$

поскольку nonmembership не является положительной premise системы.

Объект полного отрицания в Phase 4 — **автономный экспорт неоявления**:

*попытка получить отрицательный code либо named/world-absence только из того, что положительный code не появился в одном зафиксированном closure, без явной семьи продолжений, negative constructor, completeness boundary и application binding.*

Этот экспорт закрывается не риторикой, а тремя точными результатами: CBE-4 устанавливает, когда отсутствие code доказано внутри семьи; MCW-4 локализует точные контрпродолжения; BND-4 показывает, что сама семья является input и не содержится в базовом closure.

## 2. Линия Phase 0 → Phase 1 → Phase 2 → Phase 3 → Phase 4

Phase 0 потребовала bridge для перехода formal/application/goal. Phase 1 вывела различающий channel в input. Phase 2 локализовала origin среди предков. Phase 3 сохранила introduction внутри циклов и связала membership least closure с конечным support tree. Phase 4 не повторяет эти результаты: она квантифицирует по finite family extensions, локализует minimal counterextensions и исследует family-relative nonmembership.

Итоговая зависимость:

```
bridge для межтипового перехода
-> channel для различения
-> origin среди предков
-> introduction внутри цикла
-> boundary для устойчивого отсутствия
```

Каждый шаг добавляет точный объект. Negative payload не получает исключения.

## 3. Finite positive support system

Фиксируется типизированная система

$$P = (N, B, S, A, R),$$

где:

- $N$  — конечное множество exact claim-codes;
- $B$  — конечное множество protected base-ids;
- $B = S \dot{\cup} A$ :  $S, A \subseteq B$ ,  $S \cap A = \emptyset$ , и latent bases вне current/candidate partition нет;
- $R$  — конечное множество positive hyperrules;
- каждое  $r \in R$  имеет  $Prem(r) \subseteq N$ ,  $ReqBase(r) \subseteq B$ ,  $head \quad Head(r) \in N$  и  $|Prem(r)| + |ReqBase(r)| > 0$ .

Base-role и claim-role — разные координаты, даже если строки id совпали. Introduction без claim-premises допустима только при непустом  $ReqBase(r)$ . Полностью nullary rules, отрицательных premises, negation-as-failure, defeat и удаления в  $R$  нет. Это admission condition, а не вывод.

Для  $U \subseteq B$  оператор одного шага над  $X \subseteq N$ :

$$T_U(X) = X \cup \{Head(r) : r \in R, ReqBase(r) \subseteq U, Prem(r) \subseteq X\}.$$

Closure определяется как least fixed point:

$$Cl_R(U) = \mu T_U, X_0 = \emptyset.$$

Так как  $N$  конечно и оператор монотонен, итерация от  $\emptyset$  стабилизируется после конечного числа добавлений. Встраивание результата SG-LFP-3 в этот typed runtime даёт:

$$q \in Cl_R(U) \Leftrightarrow \text{для } q \text{ существует конечный derivation witness с rule-level required bases из } U.$$

Phase 4 не переименовывает этот membership в истину. Он означает только достижимость code по объявленным positive rules.

#### 4. Extension alphabet и объявленная семья

Множество  $A \subseteq B$  уже фиксировано как конечный alphabet candidate base-ids. Объявляется конечная непустая семья

$$\emptyset \in E \subseteq P(A).$$

Условие  $\emptyset \in E$  сохраняет базовый запуск без добавленного extension. Оно не утверждает, что в названном мире существует «пустое состояние».

Для каждого  $E \in E$  определяется:

$$C_E = Cl_R(S \cup E).$$

Элемент  $E$  является точным набором добавленных base-ids данного запуска. Base-id сам не входит в claim-closure; он только удовлетворяет *ReqBase*. Семья  $E$  может быть всей  $P(A)$ , строгим подмножеством или перечислением, заданным иным schema-valid контрактом. В каждом случае доказательство квантифицирует только по её членам.

Stable code-absence определяется:

$$StableAbsent_E(q) : \Leftrightarrow \forall E \in E : q \notin C_E.$$

Это meta-level predicate аудита. Он не является элементом  $N$ , пока отдельный encoding contract не введёт соответствующий code.

Максимальные admissible extensions:

$$Max(E) = \{M \in E : \nexists E \in E (M \subsetneq E)\}.$$

Конечность и непустота  $E$  гарантируют, что каждый  $E \in E$  содержится хотя бы в одном  $M \in Max(E)$ .

## 5. Две леммы, на которых держится CBE-4

### 5.1. Монотонность closure по base

Если  $U \subseteq U' \subseteq B$ , то

$$Cl_R(U) \subseteq Cl_R(U').$$

Доказательство проводится индукцией по итерациям положительного оператора. Каждое required-base условие, выполненное в  $U$ , выполнено в  $U'$ . Если claim-premises правила появились слева, они появились и справа; значит, head также появился. После стабилизации включение сохраняется.

### 5.2. Максимальное продолжение конечного элемента

Для каждого  $E \in E$  существует  $M \in Max(E)$ , такое что  $E \subseteq M$ .

Если  $E$  не максимален, выбирается строго большее допустимое extension. Строгое увеличение не может продолжаться бесконечно, поскольку  $A$  конечно. Последний элемент цепи максимален.

Обе леммы относятся к exact sets. Слова больше, полнее и максимальный не получают временного, ценностного или онтологического значения.

## 6. CBE-4: теорема максимальных продолжений

Для любого  $q \in N$  определим

$$CE_E(q) = \{E \in E : q \in C_E\}, MinCE_E(q) = Min_{\subseteq}(CE_E(q)).$$

Тогда:

$$StableAbsent_E(q) \Leftrightarrow \forall M \in Max(E) : q \notin Cl_R(S \cup M) \Leftrightarrow CE_E(q) = \emptyset \Leftrightarrow MinCE_E(q) = \emptyset.$$

Эквивалентная положительная форма:

$$\exists E \in E : q \in C_E \Leftrightarrow \exists M \in Max(E) : q \in C_M.$$

### Доказательство

Если  $q$  отсутствует во всех  $C_E$ , он отсутствует и в closures максимальных членов, поскольку каждый максимальный член принадлежит  $E$ .

Обратно, предположим, что  $q$  отсутствует в closure каждого максимального extension, но существует  $E_0 \in E$  с  $q \in C_{E_0}$ . По лемме 5.2 существует максимальный  $M \supseteq E_0$ . По монотонности

$$C_{E_0} \subseteq C_M,$$

поэтому  $q \in C_M$ , что противоречит предпосылке. Следовательно, такого  $E_0$  нет и  $StableAbsent_E(q)$  выполнен. Равенство с пустотой  $CE$  следует из определения universal; для конечной family  $CE \neq \emptyset$  ровно тогда, когда у него существует inclusion-minimal element. ▢

Теорема не использует payload  $q$ . Она одинаково действует для code, побайтно изображающих число, теорема, ракета, летит, цель, смысл, нет цели и любые другие строки.

## 7. Положительный control: конечное отсутствие действительно доказывается

Пусть

$$A = \{a, b\}, E = P(A),$$

а  $q$  не выводится ни одним rule-tree, чьи required bases входят в  $S \cup A$ . У  $E$  один максимальный член  $A$ . Если

$$q \notin Cl_R(S \cup A),$$

то по CBE-4:

$$\forall E \subseteq A : q \notin Cl_R(S \cup E).$$

Это не UNKNOWN. Это доказанное отсутствие code  $q$  во всех  $2^{|A|}$  extensions точного finite carrier. Если  $E$  перечислена полностью и её maxima проверены, перебор каждого не максимального extension не требуется.

Запрещено другое продолжение:

$q$  отсутствует во всех closures объявленной  $E$   
therefore  
названного  $q$  нет в мире

Вторая строка требует как минимум binding code к названному, scope-record, критерий полноты admissible extensions и правило экспорта. CBE-4 не содержит этих объектов.

Положительный control принципиален. Если исследование запрещало бы даже конечный вывод по полностью перечисленному carrier, оно разрушило бы собственную теорему. Жёсткая граница проходит не между «можно» и «нельзя доказывать», а между точным квантором по  $E$  и незаписанным расширением этого квантора.

## 8. MCW-4: минимальные контрпродолжения

Вернёмся к уже определённом множеству counterextensions:

$$CE_E(q) = \{E \in E : q \in C_E\}.$$

и его минимальным элементам:

$$MinCE_E(q) = \{ M \in CE_E(q) : \nexists E \in CE_E(q) (E \subsetneq M) \}.$$

### Теорема MCW-4

Для конечной объявленной семьи:

$$\neg StableAbsent_E(q) \Leftrightarrow MinCE_E(q) \neq \emptyset.$$

Кроме того, для каждого  $E \in E$ :

$$q \in C_E \Leftrightarrow \exists M \in MinCE_E(q) : M \subseteq E.$$

### Доказательство

Если *stable absence* ложно,  $CE_E(q)$  непусто. Конечное непустое частично упорядоченное множество имеет минимальный элемент. Обратно, любой *minimal counterextension* принадлежит  $CE_E(q)$ , поэтому *universal absence* ложно. Для каждого такого *member membership*  $q$  имеет конечный *derivation witness*: каждый *rule-node* перечисляет *required bases* из  $S \cup M$  и дочерние *witnesses* для *claim-premises*.

Если  $q \in C_E$ , множество *counterextensions*, содержащихся в  $E$ , непусто и конечно; у него есть минимальный элемент  $M \subseteq E$ . Такой  $M$  минимален и во всём  $CE_E(q)$ : меньший *counterextension* также содержался бы в  $E$ . Обратное направление следует из монотонности *closure*.  $\square$

MCW-4 не утверждает, что  $M$  является минимальной физической причиной. Она устанавливает минимальность набора *extension-codes* относительно точных  $E, R, S, q$ . Другой *rule-set* или другая семья могут дать другой *antichain*.

## 9. Минимальный разрушающий пример

Пусть

$$N = \{ q \}, B = \{ a \}, S = \emptyset, A = \{ a \},$$

и единственное правило  $r$  имеет вид

$$Prem(r) = \emptyset, ReqBase(r) = \{ a \}, Head(r) = q.$$

Для семьи

$$E_0 = \{ \emptyset \}$$

получается  $C_{\emptyset} = \emptyset$ , поэтому  $StableAbsent_{E_0}(q)$  истинно.

Для семьи

$$E_1 = \{ \emptyset, \{a\} \}$$

максимальным является  $\{a\}$ , а

$$q \in Cl_R(\{a\}).$$

Следовательно,

$$MinCE_{E_1}(q) = \{ \{a\} \}$$

и stable absence ложно.

Базовые  $N, B, S, R, A$  в двух случаях одинаковы. Изменился только declared boundary  $E$ . Этот конечный объект одновременно является positive control CBE-4, witness MCW-4 и разделяющей парой BND-4.

## 10. Отрицательный code не является nonmembership

Пусть  $q^+, q^- \in N$  — два различных code:

$$q^+ \neq q^-.$$

Можно объявить metadata-пару  $polar(q^+) = q^-$ , но metadata не добавляет правило вывода. Возможны четыре состояния одного closure:

| $q^+ \in C_E$ | $q^- \in C_E$ | Точный record-status                                             |
|---------------|---------------|------------------------------------------------------------------|
| да            | нет           | присутствует positive-code                                       |
| нет           | да            | присутствует negative-code                                       |
| нет           | нет           | оба code отсутствуют                                             |
| да            | да            | оба code присутствуют; конфликт только если policy так объявляет |

Поэтому:

$$q^+ \notin C_E \neq q^- \in C_E, q^- \notin C_E \neq q^+ \in C_E.$$

Если  $q^- \in C_E$ , positive support contract даёт конечный derivation witness для самого  $q^-$ . Его claim-nodes произведены rules, а каждый rule-node перечисляет required bases из  $S \cup E$ . Это **negative-origin lemma**:

отрицательная форма *payload* не освобождает *code* от положительного происхождения *membership*.

Если  $q^-$  вводится rule без claim-premises, его origin — непустой explicit *ReqBase*. Если он head правила с claim-premises, его origin — rule application с полным деревом. Если он добавлен closed-world policy после проверки nonmembership  $q^+$ , policy, boundary и результат проверки являются отдельными входами нового слоя.

## 11. Closed-world переход как новый конструктор

Правило вида

$$\frac{q^+ \notin C_E}{q^-}$$

не принадлежит positive *R*: premise проверяет отсутствие, а не membership. Оно может быть объявлено в расширенном контракте, но тогда должны быть записаны:

- extension, относительно которого вычислялось closure;
- вся семья, если требуется stable absence;
- выбранный scope ONE-EXTENSION или ALL-DECLARED-EXTENSIONS;
- критерий завершения closure;
- code отрицательного результата;
- conflict/revision policy при последующем появлении  $q^+$ .

Переименование этого конструктора в очевидно, по умолчанию, невозможно, доказано от противного ИЛИ здравый смысл не удаляет его readset.

Для полного family-level negative constructor корректной premise является не один failed search, а точная формула CBE-4. Даже после этого conclusion остаётся отрицательным code объявленного протокола. Named/world-absence не появляется автоматически.

## 12. BND-4: граница продолжений является input

Пусть core-запись

$$K = (N, B, S, A, R, q)$$

не содержит *E*. Требуется selector  $f(K)$ , который должен вернуть stable-absence status для любой допустимой объявленной семьи над тем же *A*.

### Теорема BND-4

Если существуют  $E_0, E_1 \subseteq P(A)$  с  $\emptyset \in E_i$ , для которых

$$\text{StableAbsent}_{E_0}(q) \neq \text{StableAbsent}_{E_1}(q),$$

то не существует одного selector, читающего только core  $K = (N, B, S, A, R, q)$  и не читающего  $E$ , который корректно возвращает оба статуса:

$$\nexists f : f(K) = \text{StableAbsent}_{E_0}(q) \wedge f(K) = \text{StableAbsent}_{E_1}(q).$$

Следовательно, любое корректное различие должно читать  $E$  либо эквивалентную boundary-запись.

### Доказательство

Selector получает один и тот же argument  $K$  и возвращает одно значение. Требуемые значения различны. Одно значение не равно двум различным status-codes. ▢

Пример §9 удовлетворяет предпосылке. BND-4 является конкретизацией проекционной границы Phase 1 для отрицательного status, но её новый предмет — не target вообще, а точная граница допустимых продолжений.

## 13. Исчерпывающая семья и предел слова exhaustive

Различаются три записи:

1.  $E$  перечислена без пропусков относительно собственного manifest;
2.  $E = P(A)$  для exact finite  $A$ ;
3.  $E$  охватывает все возможности названного или мира.

Первые две проверяемы: можно канонически построить  $2^{|A|}$  subsets и сравнить их с  $E$ . Третья не следует из них, поскольку  $A$  сам является declared carrier, а связь его code с названным требует binding.

Допустим строгий положительный status:

```
DECLARED-FAMILY: FINITE-AND-EXACT
POWERSSET-COVERAGE-OF-DECLARED-A: VERIFIED
CODE-ABSENCE-IN-DECLARED-FAMILY: ESTABLISHED
NAMED-ABSENCE: NOT-ESTABLISHED
WORLD-EXHAUSTIVENESS: NOT-ESTABLISHED
```

Последние две строки не обесценивают первые три. Они сохраняют доказанный квантор и запрещают слову world незаписанно расширить его.

## 14. Доказательство от противного и closed-world policy

Формальная система может содержать  $\perp$ , отрицательные code и discharge-rule. Если из assumptions вместе с  $q^+$  выводится  $\perp$ , а правило разрешает заключить  $q^-$ , negative-code имеет proof-object. Этот positive control принимается полностью.

Но остаются различными:

$$q^- \in C_E, q^+ \notin C_E, \text{StableAbsent}_E(q^+).$$

Первая формула — membership отдельного code; вторая — nonmembership в одном closure; третья — universal nonmembership по declared family. Ни одна без binding не является несуществованием названного.

Правило

$$\frac{q^+ \notin C_E}{q^-}$$

не входит в positive  $R$ : его premise проверяет отсутствие. Такой closed-world constructor допустим в расширении, но обязан читать extension, scope, boundary, termination certificate, negative-code и revision policy. Названия «очевидно», «по умолчанию» и «невозможно» не сокращают readset.

## 15. Ракета, зачем и 100/200/300 лет

Принимается полная цепь code:

```
calculation -> binding -> implementation -> controller
              -> telemetry -> flight criterion -> flight status
```

Если positive rules действительно выводят flight-status-code, он входит в closure и имеет support tree. CBE-4 не отрицает расчёт, controller или зарегистрированный успех.

Пусть exhaustive engineering-family не выводит goal-justification-code. Тогда CBE-4 доказывает его отсутствие во всей этой family. Из этого не следует «у полёта нет цели»: family могла не содержать goal-atoms, а code мог не иметь binding к чьей-либо цели. Новый goal-record создаёт новое продолжение.

Для meaning-codes  $m^+$  и  $m^-$  отдельно проверяются их membership и stable absence. Статус «оба отсутствуют» не равен  $m^-$ ; membership  $m^-$  не равен отсутствию названного смысла. Технический успех не отвечает «зачем» автоматически, но отсутствие goal-code также не доказывает бессмысленность.

Строки

100 лет  
200 лет  
300 лет  
никогда

не расширяют  $E$  без constructor. Повторный closure неизменного input даёт тот же результат. Новый sensor, rule или goal образует  $E'$ , к которой CBE-4 применяется заново. Старый verdict остаётся точным в старом score; слово «никогда» не создаёт temporal/world boundary.

## 16. Сильнейшие возражения

| Возражение                         | Точный ответ                                                                          |
|------------------------------------|---------------------------------------------------------------------------------------|
| Все варианты перебраны             | Да, absence доказано, если «все» означает exact finite $E$ ; экспорт в мир не следует |
| Maxima слишком велики              | Стоимость влияет на artifact completion, не на конечную теорему                       |
| Отрицание доказано от противного   | Принимается negative-code и trace; named absence требует binding                      |
| Closed world стандартен            | Стандартность — label; boundary различает outputs и входит в input                    |
| Ничего больше добавить нельзя      | Проверяемо для declared $A, E$ , не для незаданного будущего                          |
| Minimal counterextension — причина | Это минимальный code-set относительно $R, E$ , не causal status                       |
| Нет $q^-$ , значит есть $q^+$      | Нет: symmetric nonmembership не создаёт membership                                    |
| Мир — maximal extension            | Слово «мир» не предъявляет члена $E$                                                  |
| Узкая family бессмысленна          | Нет: её verdict точен; запрещено лишь скрывать score                                  |

## 17. Admission gates и обязательные controls

Теоремы требуют: finite exact claim carrier  $N$  и base carrier  $B = S \dot{\cup} A$ ; finite positive  $R$  с непустой суммой claim-premises и required bases; finite nonempty  $E \subseteq P(A)$  с  $\emptyset \in E$ ; least positive closure от  $\emptyset$ ; set-inclusion order;  $q \in N$ . Нарушение gate выводит объект из области, но не опровергает theorem.

Полный audit обязан включать:

- family-relative absence при отсутствии  $q$  во всех maximal closures;

- presence в одном maximal extension и отказ stable absence;
- exact  $MinCE_E(q)$ ;
- current-base-triggered случай, где  $\emptyset$  — minimal counterextension;
- независимый вывод  $q^-$  из explicit base-activated rule либо rule-chain;
- оба polar codes отсутствуют;
- оба присутствуют, а conflict-status читает policy;
- одинаковая база и разные families дают разные status;
- exact powerset control;
- application fence не повышает family verdict до world verdict.

Одни отрицательные fixtures недостаточны: checker мог бы всегда возвращать UNKNOWN.

## 18. Точные фальсификаторы

- **CBE-MAX-FALSIFIER:** все admission gates выполнены,  $q$  отсутствует во всех maximal closures, но присутствует в некотором  $C_E$ , либо эквивалентность нарушена в обратную сторону.
- **MCW-MIN-FALSIFIER:** counterextensions существуют, но minimal element отсутствует; либо объявленный minimum не выводит  $q$ ; либо имеет меньший counterextension.
- **NEGATIVE-CODE-CONFLATION:**  $q^-$  получает membership только из  $q^+ \notin C_E$ , хотя negative constructor отсутствует.
- **BOUNDARY-OMISSION:** runs имеют одну базу, разные  $E$  и разные правильные status, но процедура не читает boundary.
- **NAMED-EXPORT-SPOOF:** ABSENT-IN-DECLARED-COMPLETION-CLASS повышен до NAMED-ABSENCE-ESTABLISHED без binding и scope-rule.
- **PAYLOAD-SPOOF:** строки exhaustive, negative, world или proof принимаются как protected fields по написанию.

Первые два являются theorem-falsifiers внутри неизменного класса. Остальные поражают conformance или application, не подменяя математический контрпример.

## 19. Самоприменение

Эта рукопись — finite record. Для её exact bytes можно полностью проверить отсутствие byte-sequence. Такой verdict является положительным применением CBE-4 к предъявленному carrier.

Из отсутствия строки в файле не следует отсутствие названного. Из отсутствия внешней ссылки не следует историческое отсутствие предшественника. Из отсутствия контрпримера среди fixtures не следует его отсутствие в любом классе. Каждый квантор требует собственной family.

Заголовок без подзаголовка получает UNBOUNDED-STANDALONE. Точное reading задают CBE-4, MCW-4, negative-origin lemma и BND-4. Если позднее создан новый файл, он является новым extension-record, а не материалом, тайно прочитанным ранним audit.

## 20. Claim/status и источники

C-PUBLIC-TITLE

expected\_status: UNBOUNDED-STANDALONE

C-THEOREM-CBE-4

C-THEOREM-MCW-4

C-THEOREM-BND-4

expected\_status: TEXTUAL-THEOREM-IN-DECLARED-METATHEORY

semantic\_status: NOT-MACHINE-VERIFIED

C-APPLICATION-NAMED-ABSENCE

expected\_status: NOT-ESTABLISHED-WITHOUT-BINDING-AND-BOUNDARY

Manuscript-метки и executable fields не склеиваются:

| Reading рукописи                                 | Exact executable representation                                                                 |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------|
| record прошёл runtime validation и полный анализ | overall VALID-EXTENSION-ABSENCE-ANALYSIS                                                        |
| current claim присутствует                       | NOT-AN-ABSENCE-CURRENTLY-PRESENT                                                                |
| COUNTEREXTENSION-FOUND при local nonmembership   | CURRENTLY-ABSENT-ENABLED-BY-DECLARED-EXTENSION +<br>непустой minimal_enabling_counterextensions |
| ABSENT-IN-DECLARED-COMPLETION-CLASS              | STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS                                                         |

Overall status не является absence verdict; per-claim absence\_class не является новым claim-code. Во всех случаях runtime сохраняет fences DECLARED-FINITE-EXTENSIONS-ONLY-NOT-FUTURE-EXHAUSTIVENESS-ATTESTED и POSITIVE-NEGATIVE-CLAIM-CODES-ONLY-NOT-WORLD-POLARITY-VERIFIED.

Ни один внешний автор, школа или термин не является premise. Объекты и доказательства предъявлены внутри рукописи.

PREDECESSORS-AS-PREMISES: NONE

INTERNAL-PHASE-CONTRIBUTION: CBE-4 + MCW-4 + BND-4

EXTERNAL-COMPARISON: NOT-USED-AS-FOUNDATION

PRIORITY-STATUS: NO-PRIORITY-CLAIM-IN-THIS-RECORD

Последний status не сообщает, существуют ли совпадения. Он не позволяет отсутствию comparative record стать доказательством исторического отсутствия.

## 21. Граница притязаний

### Установлено

1. Positive closure монотонно по base, а каждый член finite family содержится в maximal extension.

2. CBE-4: absence во всей family эквивалентно absence во всех maximal closures.
3. Exact finite family позволяет полностью доказать code-absence в своём домене.
4. MCW-4: stable absence ложно ровно при наличии minimal counterextension; каждый deriving extension содержит одно из них.
5. Nonmembership  $q^+$  не является membership  $q^-$ ; выведенный negative-code имеет собственный origin и support tree.
6. BND-4: boundary входит в readset, если families меняют status при одной базе.
7. Family-relative absence не экспортируется в named/world absence без нового binding и scope.

### Не установлено

- существование или отсутствие named object;
- что code предъявляет объект, событие, причину, цель или смысл;
- что  $A, E$  исчерпывают мир или будущее;
- истинность negative-code и невозможность polar conflict;
- правильность closed-world, default, defeat или revision policy;
- охват arbitrary nonmonotone либо infinite systems;
- причинность minimal counterextension;
- что полёт отвечает на «зачем» или отсутствие goal-code означает отсутствие цели;
- ложность математических proofs;
- историческое происхождение математики;
- мировой приоритет Phase 4.

Эта граница сохраняет сильный конечный квантор и отсекает только непрочитанный domain.

## 22. Итог

Положительный code может иметь корректный support tree. Negative-code может иметь корректный proof-object. Exact finite family может доказать, что code не появляется ни в одном допустимом closure. Всё это принято.

Но различны:

$$q^+ \notin C_E, \text{StableAbsent}_E(q^+), q^- \in C_E.$$

Максимальные продолжения дают полный family-test. Минимальные counterextensions показывают, какой exact input разрушает absence. BND-4 показывает, почему boundary не получается повторным вычислением базы.

Опровергнуто притязание:

*Непоявление positive-code в одной записи само не создаёт negative-code и не доказывает отсутствие названного за declared family.*

Сохранён положительный результат:

*Если finite family объявлена точно и code отсутствует в closure каждого максимального члена, его отсутствие доказано во всей family.*

Ракета может быть записана как полетевшая; goal-code и negative goal-code могут быть добавлены; 100/200/300 лет могут стать новыми records. Каждый новый материал меняет input там, где введён. Непоявление строки не переписывает собственное происхождение.

*Отсутствие в объявленном доказано объявленным.*

*Необъявленное не стало отсутствующим.*

---

## **Статус источников**

Работа продолжает внутреннюю авторскую линию beforeword и Phase 0–3. Внешние сравнительные источники в посылки не включены; global novelty и historical priority этим не устанавливаются.

## 01. ТЕОРЕМА ПРОИСХОЖДЕНИЯ ОТРИЦАТЕЛЬНОГО СТАТУСА

### НЕ НАЙДЕНО НЕ УСТАНОВЛИВАЕТ ОТСУТСТВИЕ

#### 0. Нормативная область и нулевая уступка словам

Этот файл задаёт конечное ядро Phase 4. Его объект — переход от результата поиска к отрицательному status-code. Рассматривается не всё, что может быть названо словами нет, отсутствует, невозможно ИЛИ не существует, а только точный класс конечных расширений одной записи.

Строки

NOT-FOUND  
ABSENT  
COMPLETE  
ALL-CHECKED  
младенец  
число

остаются строками. Ни одна из них не предъявляет названное. В частности, наличие строки ABSENT в payload не является отсутствием чего-либо; строка COMPLETE не завершает класс продолжений; строка младенец не предъявляет младенца. Защищённые поля ниже вводят только отношения между exact-record codes.

Публичная строка Не найдено не устанавливает отсутствие без области имеет status-class UNBOUNDED-STANDALONE. Теоремный статус имеют только CBE-4 и MCW-4 при явно перечисленных предпосылках. Их вывод ABSENT-IN-DECLARED-COMPLETION-CLASS не сокращается до неограниченного ABSENT.

#### 1. Конечный типизированный completion-boundary contract

##### 1.1. Два защищённых сорта

Фиксируется immutable input-record  $r_0$ . Пусть  $N$  — конечное непустое множество claim-codes, а  $B$  — конечное множество base-id. Это разные protected namespaces; в абстрактном вложении используются теги CLAIM и BASE, поэтому

$$N \cap B = \emptyset.$$

Пусть  $S \subseteq B$  — current base-set. Runtime допускает как candidates все остальные bases:

$$A = B \setminus S, B = S \dot{\cup} A.$$

Фиксируется query-code  $q \in N$ . Добавление  $\Delta \subseteq A$  меняет активный base-set на

$$U_{\Delta} = S \cup \Delta.$$

Exact append  $r_0 \oplus \Delta$  сохраняет protected fields  $r_0$  и добавляет canonical set base-id. Payload не создаёт ни base-id, ни claim-code.

## 1.2. Допустимая completion-family

Boundary задаёт точное конечное семейство

$$\emptyset \in E \subseteq P(A).$$

Элемент  $\Delta \in E$  является admissible addition-set только относительно этого record. Семья не обязана быть downward-closed либо равной  $P(A)$ .

$$Max(E) = \{M \in E : \nexists \Gamma \in E (M \subsetneq \Gamma)\}.$$

Конечность и непустота гарантируют: каждый  $\Delta \in E$  содержится в некотором  $M \in Max(E)$ . Слово maximal не означает физическую или историческую полноту.

## 1.3. Exact input map

Нормативный record включает:

- bases[] как каталог  $B$  и current\_base\_ids[] как  $S$ ;
- derived candidate-set  $A = B \setminus S$ ;
- claims[] как  $N$ ;
- полный frozen inferences[] как  $R$ ;
- admissible\_extensions[] как  $E$ , включая ровно один  $\emptyset$ ;
- query выбирает различные  $q^+, q^- \in N$ ; все теоремы ниже инстанцируются отдельно при  $q = q^+$  и  $q = q^-$ ;
- canonicalization и policy least-fixed-point-positive-membership-v1.

Executable schema специализирует конечное ядро границами  $|B| \leq 12$ ,  $2 \leq |N| \leq 18$ ,  $|R| \leq 36$ ,  $1 \leq |E| \leq 64$ . Эти engineering bounds не участвуют в доказательствах, кроме конечности.

Checker выводит maxima из полной  $E$ . Payload complete и поле с похожим именем coverage не создают.

## 2. Положительное типизированное замыкание

### 2.1. Rules с claim- и base-premises

Каждое  $r \in R$  имеет защищённые поля

$$Prem(r) \subseteq N, ReqBase(r) \subseteq B, Head(r) \in N,$$

и domain-condition

$$|Prem(r)| + |ReqBase(r)| > 0.$$

Claim-premises могут быть пусты, если rule требует хотя бы один base-id. Поэтому base-only introduction имеет форму

$$Prem(r) = \emptyset, ReqBase(r) \neq \emptyset.$$

Nullary rule без claim- и base-premises отвергается executable checker: overall status INVALID-EXTENSION-ABSENCE-RECORD, reason NULLARY\_INFERENCE\_REQUIRES\_EXPLICIT\_BASE. Такой rule находится вне contract. Negation-as-failure, deletion, retraction, defeat и изменение rule-set между extensions не входят в executable core.

### 2.2. Least closure и persistence

Для активного  $U \subseteq B$ :

$$T_U(X) = X \cup \{Head(r) : r \in R, ReqBase(r) \subseteq U, Prem(r) \subseteq X\}.$$

Положим  $X_U^0 = \emptyset$ ,  $X_U^{n+1} = T_U(X_U^n)$  и

$$C_\Delta = \mu T_{U_\Delta} = \bigcup_{n \geq 0} X_{U_\Delta}^n.$$

Witness predicate:

$$W_q(\Delta) \Leftrightarrow q \in C_\Delta.$$

Если  $\Delta, \Gamma \in E$  и  $\Delta \subseteq \Gamma$ , то  $U_\Delta \subseteq U_\Gamma$ . Индукция по  $n$  даёт

$$X_{U_\Delta}^n \subseteq X_{U_\Gamma}^n, C_\Delta \subseteq C_\Gamma,$$

следовательно,

$$W_q(\Delta) \Rightarrow W_q(\Gamma).$$

Это membership claim-code по exact base requirements. Оно не предъявляет референт, событие, количество, объект или внешнюю истинность. Retractable/effective-state target снимает W-PERSIST и требует другого контракта.

---

### 3. Три уровня отрицательного результата

#### 3.1. Процедурная неудача поиска

Строка

NO-WITNESS-RETURNED

означает только, что запуск не вернул witness-code. Timeout, ограничение глубины или неполный parser могут дать эту строку при  $q \in C_{\emptyset}$ . Поэтому она не является даже локальным отрицанием.

#### 3.2. Локальное исчерпывающее NOT-FOUND

После точного вычисления least closure текущей записи допускается status

$$LNF(q) \Leftrightarrow \neg W_q(\emptyset).$$

Он устанавливает nonmembership только при current bases  $S$ . Он не квантифицирует по строгим  $\Delta \in E$ .

#### 3.3. Отсутствие в объявленном классе завершений

Boundary-relative absence определяется независимо:

$$StableAbsent_E(q) \Leftrightarrow \forall \Delta \in E : \neg W_q(\Delta).$$

Его единственный нормативный status-code:

ABSENT-IN-DECLARED-COMPLETION-CLASS

Даже этот status не означает, что отсутствует названное строкой  $q$ . Он означает STABLE-ABSENCE для exact  $r_0, N, B, S, A, E, R, q$  и fixed policy.

---

### 4. Полярностное разделение

Положительная и отрицательная стороны имеют разную форму.

Если

$$W_q(\emptyset),$$

to finite typed derivation witness удостоверяет  $q \in C_{\emptyset}$ . Поскольку  $\emptyset \in E$ , сразу следует

$$\neg \text{StableAbsent}_E(q).$$

Если же

$$\neg W_q(\emptyset),$$

получен только LNF. Stable absence требует универсальной проверки completion-family.

Итак, всегда верны импликации

$$W_q(\emptyset) \Rightarrow \neg \text{StableAbsent}_E(q),$$

$$\text{StableAbsent}_E(q) \Rightarrow \text{LNF}(q).$$

Для polarity audit допускаются два защищённых и различных claim-code

$$q^+, q^- \in N, q^+ \neq q^-.$$

Непринадлежность положительного code является meta-условием, а не принадлежностью отрицательного code:

$$\neg W_{q^+}(\Delta) \not\Rightarrow W_{q^-}(\Delta).$$

Для  $q^-$  также нужен finite typed derivation по frozen rules. Возможны все четыре exact состояния:

| $W_{q^+}(\Delta)$ | $W_{q^-}(\Delta)$ | Что установлено                            |
|-------------------|-------------------|--------------------------------------------|
| 1                 | 0                 | positive code входит в closure             |
| 0                 | 1                 | negative code входит в closure             |
| 0                 | 0                 | оба codes не входят                        |
| 1                 | 1                 | оба codes входят; conflict-policy отдельно |

Ни одна строка таблицы не предъявляет названное. Closed-world переход

$$\neg W_{q^+}(\Delta) \Rightarrow W_{q^-}(\Delta)$$

является новым policy-constructor. В positive core он не возникает из молчания. Даже CBE-4 для  $q^+$  устанавливает family-relative nonmembership, а не автоматически membership  $q^-$ .

Но обратная импликация

$$LNF(q) \Rightarrow StableAbsent_E(q)$$

не является правилом Phase 4. Она допустима только после проверки эквивалентных условий CBE-4.

Повторение LNF сто, двести или триста раз не перечисляет ветви  $E$  и не создаёт boundary.

## 5. CBE-4: теорема эквивалентности границы завершения

### 5.1. Counterextension sets

Определим множество witness-bearing completions:

$$CE_E(q) = \{ \Delta \in E : W_q(\Delta) \}.$$

Eго inclusion-minimal antichain:

$$MinCE_E(q) = \{ \Delta \in CE_E(q) : \nexists \Gamma \in CE_E(q) (\Gamma \subsetneq \Delta) \}.$$

### 5.2. Теорема CBE-4

Для каждого finite admissible contract разделов 1–2 эквивалентны:

- (C1)  $StableAbsent_E(q)$ ;
- (C2)  $\forall M \in Max(E) : \neg W_q(M)$ ;
- (C3)  $CE_E(q) = \emptyset$ ;
- (C4)  $MinCE_E(q) = \emptyset$ .

Тем самым maximal completions являются достаточным верхним фронтиром проверки, а пустота minimal counterextension antichain — эквивалентным отрицательным сертификатом внутри finite boundary.

### 5.3. Доказательство

Эквивалентность C1 и C3 следует из определений.

Из C1 следует C2, поскольку  $Max(E) \subseteq E$ .

Пусть C2 выполнено, но C1 ложно. Тогда  $W_q(\Delta)$  для некоторого  $\Delta \in E$ . Конечность даёт maximal  $M \in E$  с  $\Delta \subseteq M$ . По W-PERSIST  $W_q(M)$ , вопреки C2.

Если CE пусто, MinCE пусто. Обратно, если CE непусто, конечное частично упорядоченное множество CE имеет inclusion-minimal элемент. Он входит в MinCE. Поэтому пустота MinCE влечёт пустоту CE. Все четыре условия эквивалентны.  $\square$

## 5.4. Что CBE-4 не говорит

CBE-4 не утверждает, что  $E$  исчерпывает события, предметы, людей, будущие записи или всё употребление слова математика. Новый base-id вне  $B$  либо addition вне  $E$  показывает лишь, что boundary не покрывал новое применение.

## 6. MCW-4: теорема минимального контррасширения

### 6.1. Теорема

$$\neg \text{StableAbsent}_E(q) \Leftrightarrow \text{MinCE}_E(q) \neq \emptyset.$$

Для каждого  $\Gamma \in E$ :

$$W_q(\Gamma) \Leftrightarrow \exists \Delta^* \in \text{MinCE}_E(q) : \Delta^* \subseteq \Gamma.$$

Для каждого  $\Delta^* \in \text{MinCE}_E(q)$  существует finite typed derivation witness корня  $q$ . Claim-node указывает producer rule, его claim-children точно соответствуют  $\text{Prem}(r)$ , а  $\text{ReqBase}(r) \subseteq S \cup \Delta^*$ . Base-id не становится claim-leaf.

### 6.2. Доказательство

MCW-4A следует из CBE-4. Для MCW-4B пусть  $W_q(\Gamma)$ . Конечное непустое множество counterextensions, содержащихся в  $\Gamma$ , имеет minimal  $\Delta^* \subseteq \Gamma$ ; меньший глобальный counterextension также содержался бы в  $\Gamma$ . Обратно,  $\Delta^* \subseteq \Gamma$  и W-PERSIST дают  $W_q(\Gamma)$ .

Derivation witness строится индукцией по first birth round claim-code в итерации  $T_{S \cup \Delta^*}$ . Producer каждого noninitial claim имеет premises меньших birth rounds. В нижнем узле claim-premises пусты, но  $\text{ReqBase}(r) \neq \emptyset$  по NONNULL-TYPED. Поэтому recursion конечна.  $\square$

### 6.3. Смысл минимальности

Если дополнительно  $LNF(q)$ , то каждый  $\Delta^* \in \text{MinCE}_E(q)$  непуст:  $\Delta^* = \emptyset$  означало бы  $W_q(\emptyset)$ . Для каждого proper  $\Gamma \in E$ :

$$\Gamma \subsetneq \Delta^* \Rightarrow \neg W_q(\Gamma).$$

$\Delta^*$  — минимальный exact bundle добавленных base-id среди members  $E$ , не обязательно singleton или единственный. Он не является минимальной физической причиной.

MinCE образует antichain. Один counterextension опровергает stable absence; полный список требует full-family pass.

## 6.4. Gate-cut corollary

Пусть  $D \subseteq A$  — protected set запрещённых extension-atoms, а новый boundary допускает ровно

$$E_D = \{ \Delta \in E : \Delta \cap D = \emptyset \}.$$

Отсутствие witness во всём  $E_D$  эквивалентно условию

$$\forall \Delta^* \in \text{MinCE}_E(q) : D \cap \Delta^* \neq \emptyset.$$

Действительно, surviving minimal counterextension допустимо и несёт witness. Если  $D$  пересекает каждое minimal counterextension, MCW-4 запрещает surviving witness-bearing extension. Это theorem о coordinate gate, не оправдание запрета  $D$ .

---

## 7. Completion boundary является различающим input

### 7.1. Boundary-sensitivity pair

Пусть

$$N = \{ q \}, B = A = \{ a \}, S = \emptyset,$$

а frozen rule  $r_a$  имеет

$$\text{Prem}(r_a) = \emptyset, \text{ReqBase}(r_a) = \{ a \}, \text{Head}(r_a) = q.$$

Рассмотрим две completion-families:

$$E_0 = \{ \emptyset \}, E_1 = \{ \emptyset, \{ a \} \}.$$

Typed core  $K = (N, B, S, A, R, q)$ , current closure и LNF совпадают. Но

$$\text{StableAbsent}_{E_0}(q) = 1, \text{StableAbsent}_{E_1}(q) = 0.$$

Поэтому никакая функция только typed core и local status не выдаёт оба ответа:

$$\nexists F : \text{StableAbsent}_{E_i}(q) = F(K, \text{LNF}(q)) (i = 0, 1).$$

Различие находится в  $E_i$ . Boundary обязан входить в полный readset.

### 7.2. Доказательство

Аргументы  $F$  в двух instances равны, следовательно, outputs  $F$  равны. Требуемые status-codes различны. Противоречие.  $\square$

Boundary может быть внесён schema или horizon-rule. Он меняет contract, но не удостоверяет, что названное словом всё исчерпано.

---

## 8. Admission gates отрицательного status

Status ABSENT-IN-DECLARED-COMPLETION-CLASS допускается только после последовательного прохождения gates.

### Gate 0 — EXACT-STRUCTURE

- $N, B, S, A, E, R, q$  конечны;
- $N \cap B = \emptyset, B = S \dot{\cup} A$ ;
- $\emptyset \in E \subseteq P(A)$ ;
- неизвестные, повторные и reintroduced current base-id отвергаются.

### Gate 1 — POSITIVE-PERSISTENCE

- extensions добавляют только base-id из  $A$ ;
- rule-set frozen и positive;
- $|Prem(r)| + |ReqBase(r)| > 0$ ;
- нет retraction, defeat или negation-as-failure, влияющих на  $W$ .

### Gate 2 — LOCAL-EXHAUSTION

Least closure  $C_\emptyset$  вычислено полностью. Timeout или bounded search не получает LNF.

### Gate 3 — COMPLETION-COVERAGE

Checker вычисляет  $C_\Delta$  для каждого  $\Delta \in E$ . Maximal pass достаточен для stable absence; exact MinCE требует full-family pass.

### Gate 4 — BOUNDARY-IN-READSET

artifact\_sha256 связывает полный input, включая  $B, S, A, E, N, R, q$ , policy и payload. Payload complete остаётся инертным.

### Gate 5 — POLARITY-VERDICT

Теоремная метка

ABSENT-IN-DECLARED-COMPLETION-CLASS

соответствует runtime-полям:

```
status = VALID-EXTENSION-ABSENCE-ANALYSIS
query_analysis.<report>.current_present = false
```

```
query_analysis.<report>.absence_class =
  STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS
query_analysis.<report>.minimal_enabling_counterextensions = []
```

Если MinCE непусто, absence\_class равен CURRENTLY-ABSENT-ENABLED-BY-DECLARED-EXTENSION либо, при current presence, NOT-AN-ABSENCE-CURRENTLY-PRESENT. В последнем случае  $MinCE_E(q) = \{\emptyset\}$ , поэтому report обязан сохранить empty extension и его witness. Поле MinCE содержит exact minimal additions и typed derivation witnesses.

<report> ОЗНАЧАЕТ positive\_report ИЛИ negative\_report; CBE-4 применяется к каждому claim-code отдельно.

## Gate 6 — SEMANTIC-FENCE

Ни один result-code не переименовывается во внешнюю истину, физическое отсутствие, невозможность будущего, отсутствие смысла или отсутствие названного. Admission такого code как premise следующего calculus является новым protected input. Строка ABSENT внутри payload не проходит gates.

Runtime сохраняет semantic\_status: NOT-MACHINE-VERIFIED и fences DECLARED-FINITE-EXTENSIONS-ONLY-NOT-FUTURE-EXHAUSTIVENESS-ATTESTED, POSITIVE-NEGATIVE-CLAIM-CODES-ONLY-NOT-WORLD-POLARITY-VERIFIED.

## 9. Положительные controls

### 9.1. Закрытый singleton

$$B = S = A = \emptyset, E = \{\emptyset\}, R = \emptyset, q \notin C_{\emptyset}.$$

Единственный maximal completion не несёт  $q$ ; MinCE пусто. Результат — ABSENT-IN-DECLARED-COMPLETION-CLASS, не внешнее отсутствие.

### 9.2. Один минимальный counterextension

$$B = A = \{a\}, S = \emptyset, E = P(A).$$

Rule  $r_a$  имеет empty claim-premises,  $ReqBase(r_a) = \{a\}$  и head  $q$ . Тогда

$$q \notin C_{\emptyset}, q \in C_{\{a\}}, MinCE_E(q) = \{\{a\}\}.$$

### 9.3. Несовместимые maxima

$$B = A = \{a, b\}, S = \emptyset, E = \{\emptyset, \{a\}, \{b\}\}.$$

Пусть base-only rule требует  $a$  и выводит  $q$ ; rule с  $b$  нет. Оба singleton maximal. Проверка только  $\{b\}$  была бы omission.

#### 9.4. Совместный witness-bundle

Пусть  $B = A = \{a, b\}$ ,  $S = \emptyset$ , а frozen rule имеет

$$Prem(r) = \emptyset, ReqBase(r) = \{a, b\}, Head(r) = q.$$

При  $E = P(\{a, b\})$ :

$$MinCE_E(q) = \{\{a, b\}\}.$$

Ни  $a$ , ни  $b$  отдельно не является достаточным.

#### 9.5. Payload spoof

Base payload может содержать полные строки

ABSENT  
COMPLETE  
all maximal extensions checked

при неизменных protected  $B, S, A, E, N, R$ . CBE-4 verdict не меняется.

#### 9.6. Cycle без base-only входа

Rule с  $Prem(r) = \{q\}$ ,  $ReqBase(r) = \emptyset$ ,  $Head(r) = q$  не помещает  $q$  в least closure. Cycle без base-only входа остаётся пустым. Greatest/nonleast policy меняет contract.

### 10. Точные falsifiers и условия снятия

#### 10.1. CBE-MAX-FALSIFIER

CBE-4 поражён внутри неизменного finite positive contract, если одновременно:

1. каждый  $M \in Max(E)$  удовлетворяет  $\neg W_q(M)$ ;
2. существует  $\Delta \in E$  с  $W_q(\Delta)$ ;
3. W-PERSIST, typed rules, конечность и exact maxima сохранены.

Если найденный maximal list просто неполон, это implementation defect, а не математический falsifier.

#### 10.2. CBE-MIN-EMPTY-FALSIFIER

Предъявить finite exact CE, которое непусто, но не имеет inclusion-minimal элемента. При обычном конечном включении такой объект невозможен. Бесконечный или некорректно канонизированный input снимает domain-premise.

### 10.3. MCW-MINIMALITY-FALSIFIER

Checker объявляет  $\Delta^*$  minimal, хотя существует  $\Gamma \in E$  с

$$\Gamma \subsetneq \Delta^* \wedge W_q(\Gamma).$$

Либо  $\neg \text{StableAbsent}_E(q)$ , но полный computed MinCE пуст.

### 10.4. MCW-UPWARD-COVER-FALSIFIER

Все premises MCW-4 сохранены, но witness-bearing  $\Gamma \in E$  не содержит  $\Delta^* \in \text{MinCE}_E(q)$ , либо допустимый superset minimal counterextension не несёт  $q$ .

### 10.5. LOCAL-SEARCH-FALSIFIER

Instance получил LNF, хотя exact recomputation даёт  $q \in C_\emptyset$ . Timeout не является LNF.

### 10.6. Не являются theorem-falsifiers

- base-id вне  $B$  либо addition вне frozen  $E$ ;
- изменение query, typed rules, current bases, canonicalization или policy;
- retractable witness либо бесконечный класс без cofinal maxima;
- payload ABSENT, рассказ вне declared relation или другой calculus.

Такие случаи создают новый contract либо показывают ошибку применения прежнего boundary. Чтобы стать theorem-falsifier, объект обязан сохранить все посылки и нарушить точную эквивалентность.

## 11. Абстрактное расширение и его обязательные условия

Пусть  $(E, \leq)$  — непустой extension-poset с current element. Если  $W$  upward-closed, maxima cofinal

$$\forall e \in E \exists m \in \text{Max}(E) : e \leq m.$$

и witness-bearing subset well-founded вниз, эквивалентности CBE-4 сохраняются. Без cofinal maxima проверка maxima может быть вакуумной или пропустить бесконечную ветвь; без persistence промежуточный witness может исчезнуть; без well-foundedness CE может не иметь minimal элемента.

В executable core конечность  $E \subseteq P(A)$  обеспечивает cofinal maxima и minimal counterextensions, а раздел 2 — W-PERSIST.

## 12. Самоприменение и граница результата

Настоящий файл также является record. Его строки CBE-4, MCW-4, доказательство, falsifier и точный не присваивают себе theorem-status. Такой status допустим только относительно предъявленных определений, правил метатеории и отдельного proof/read contract.

Если файл не найден при поиске каталога, из этого не следует его отсутствие вне exact search domain. Если файл найден, найден path/byte-record по процедуре; это не предъявляет названное содержанием файла. Если manifest перечисляет все файлы закрытого package и checker исчерпывающе проверяет его, отсутствие path может быть установлено только как ABSENT-IN-DECLARED-PACKAGE-MANIFEST.

Phase 4 устанавливает positive persistence, CBE-4, MCW-4 и необходимость completion boundary в readset. Local exhaustive NOT-FOUND не является stable absence; при LNF и  $\neg \text{StableAbsent}_E(q)$  minimal counterextensions непусты. Все результаты остаются отношениями codes.

Phase 4 не устанавливает:

- что названного объекта, события, количества, человека или смысла нет;
- что  $E$  исчерпывает непредъявленное или новый record невозможен;
- что любое доказательство отрицания неверно либо checker-output внешне истинен.

Максимальная точная строка:

*Неудача поиска относится к выполненному поиску. Отсутствие относится к заранее объявленному и проверенному классу завершений. Чтобы перейти от первого ко второму, граница завершения должна войти в readset; её имя не предъявляет полноту.*

## CWC-4: сертификат полноты конечного класса дополнений

### 0. Нормативный статус

CWC-4 — нормативный идентификатор **Completion Witness Certificate** для Phase 4: «отсутствие записи ≠ запись отсутствия».

Контракт проверяет одно ограниченное утверждение отдельно для каждого из двух query claim-codes: выбранный code не входит в least closure ни при одном дополнении из явно перечисленного конечного класса. ABSENT-IN-DECLARED-COMPLETION-CLASS не означает отсутствия объекта, события, свойства, причины, значения, смысла или возможности вне записи. Все термины ниже — имена полей и отношений над code-records; чтение не устанавливает их референт.

Основной разрез (CWC-FENCE):

*не предъявлен witness*  
*/ code не найден локально*  
*/ code отсутствует во всём объявленном классе дополнений*

Третья строка сильнее второй только внутри exact family. Переход к «нигде» или «не существует» запрещён.

### 1. Три несовпадающих статуса

#### 1.1. NO-WITNESS-RETURNED

NO-WITNESS-RETURNED фиксирует только отсутствие принимаемого witness в response. Причиной может быть лимит, ошибка, неполный readset или недостижимость. Status не утверждает ни

$$q \in \mu T_s,$$

ни

$$q \notin \mu T_s.$$

Пустое поле, timeout и строка «ничего не найдено» не являются отрицательным certificate.

#### 1.2. LOCALLY-NOT-FOUND

LOCALLY-NOT-FOUND имеет точный смысл

$$q \notin \mu T_s.$$

Проверяется только  $S$ , то есть  $E = \emptyset$ . Нужна завершённая LFP-проверка; о разрешённых дополнениях status не говорит.

### 1.3. ABSENT-IN-DECLARED-COMPLETION-CLASS

ABSENT-IN-DECLARED-COMPLETION-CLASS имеет точный manuscript-смысл

$$\forall E \in E \ q \notin \mu T_{S \cup E},$$

где  $E$  — finite extensional family внутри одного input, связанного artifact\_sha256. В runtime этому manuscript-label точно соответствует STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS. Из  $\emptyset \in E$  следует (ADCC)  $\Rightarrow$  (LOCAL-NF); обратно — нет.

NO-WITNESS-RETURNED не располагается на этой шкале: это diagnosis response, а не membership-теорема.

## 2. Конечный completion-contract

### 2.1. Две типизированные области

Пусть  $N$  — конечное множество claim\_id как минимум с двумя различными элементами: universe claim-codes. Пусть  $B$  — конечное множество base\_id: отдельный каталог оснований. Типы не подменяют друг друга. В частности, base\_id не является claim-code и никогда не помещается в claim closure.

Текущий набор оснований есть

$$S \subseteq B,$$

а кандидатная область дополнений определяется точно как

$$A = B \setminus S, B = S \sqcup A.$$

Буква  $A$  означает только declared candidate-base domain. Она не утверждает доступность чего-либо вне записи.

Пусть

$$\emptyset \neq E \subseteq P(A), \emptyset \in E,$$

— явно перечисленное конечное семейство допустимых дополнений. Каждый  $E \in E$  является набором дополнительных base\_id, а не claim-codes. Семейство не обязано быть downward closed, upward closed или равным  $P(A)$ . Любое из этих свойств должно быть отдельно проверено, если оно заявлено.

Равенство payload-строк не склеивает protected ids. Включение в  $S$  или  $A$  задаётся только защищённым полем.

## 2.2. Положительные ненулевые hyperrules

Пусть  $R$  — конечное множество rules

$$r = \{ id_r, Prem(r), ReqBase(r), Head(r) \},$$

для которых

$$Prem(r) \subseteq N, ReqBase(r) \subseteq B, Head(r) \in N, |Prem(r)| + |ReqBase(r)| > 0.$$

Rules положительны: только inclusion premises и требуемых оснований, без отрицания и удаления. Пустой Prem допустим, если ReqBase непуст. Rule с обоими пустыми наборами является структурно недействительной записью: runtime возвращает INVALID-EXTENSION-ABSENCE-RECORD с reason-code NULLARY\_INFERENCE\_REQUIRES\_EXPLICIT\_BASE.

Для любого  $U \subseteq B$  задаётся оператор над claim-codes

$$T_U(X) = X \cup \{ Head(r) : r \in R, ReqBase(r) \subseteq U, Prem(r) \subseteq X \}.$$

Его least fixed point

$$\mu T_U = \bigcup_{i \geq 0} X_U^i, X_U^0 = \emptyset, X_U^{i+1} = T_U(X_U^i),$$

стабилизируется, поскольку  $N$  конечно. Начальный claim-set всегда пуст: основания в  $U$  лишь разрешают правила и сами не становятся элементами  $X$ .

## 2.3. Completion predicate

Для fixed query  $q \in N$  и любого  $E \subseteq A$  определяется булев предикат

$$W_q(E) : \Leftrightarrow q \in \mu T_{S \cup E}.$$

$w$  — только membership в closure, не истинность, существование или цель.

## 3. Монотонность дополнения

### Лемма CWC-MON

Для любых  $E, F \subseteq A$ :

$$E \subseteq F \Rightarrow \mu T_{S \cup E} \subseteq \mu T_{S \cup F}.$$

Следовательно,

$$E \subseteq F \wedge W_q(E) \Rightarrow W_q(F).$$

### Доказательство CWC-MON

Индукцией по  $i$ :

$$X_{S \cup E}^i \subseteq X_{S \cup F}^i.$$

База пуста. Если правило применимо слева, то его `required_base_ids` входят также в  $S \cup F$ , а его `premise claim-codes` уже выведены справа по индукционному предположению. Объединение раундов даёт результат. ▢

При отрицательных `premises` или удалении `code` лемма неприменима.

## 4. Максимальный фронтير семейства

Определяется inclusion-maximal frontier:

$$Max(E) = \{ M \in E : \nexists F \in E (M \subsetneq F) \}.$$

Конечность даёт для каждого  $E \in E$  maximal  $M \supseteq E$ ; промежуточные subsets не обязаны входить в family.

### Теорема MAX-FRONTIER-4

$$\forall E \in E \neg W_q(E) \Leftrightarrow \forall M \in Max(E) \neg W_q(M)$$

### Доказательство MAX-FRONTIER-4

Прямое направление следует из  $Max(E) \subseteq E$ . Если обратно некоторый  $E$  истинен, maximal  $M \supseteq E$  истинен по CWC-MON — противоречие. ▢

Для (ADCC) достаточна exhaustive evaluation  $Max(\mathcal{E})$ , но не выборка.

## 5. Минимальные контрдополнения

Множество истинных дополнений внутри объявленного класса:

$$CE_q(E) = \{ E \in E : W_q(E) \}.$$

Ero inclusion-minimal frontier:

$$MinCE_q(E) = \{ E \in CE_q(E) : \nexists F \in CE_q(E) (F \subsetneq E) \}.$$

Counterextension — контрпример к (ADCC), не к внешнему объекту.

## Свойства MinCE

1.  $MinCE_q(E)$  — antichain по  $\subseteq$ .
2. Она пуста ровно тогда, когда выполняется ABSENT-IN-DECLARED-COMPLETION-CLASS.
3. Если  $W_q(\emptyset)$ , то

$$MinCE_q(E) = \{ \emptyset \}.$$

4. Для каждого  $E \in CE_q(E)$  существует  $M \in MinCE_q(E)$  с  $M \subseteq E$ .
5. Положительный maximal member достаточен, чтобы опровергнуть (ADCC), но не определяет exact MinCE.

## СВЕ-4: граничная эквивалентность

Для выбранного claim-code  $q$  определяется

$$StableAbsent_E(q) : \Leftrightarrow \forall E \in E \neg W_q(E).$$

Тогда CWC-MON и конечность family дают полный ряд эквивалентностей:

$$StableAbsent_E(q) \Leftrightarrow \forall M \in Max(E) \neg W_q(M) \Leftrightarrow CE_q(E) = \emptyset \Leftrightarrow MinCE_q(E) = \emptyset.$$

Первое выражение есть manuscript ABSENT-IN-DECLARED-COMPLETION-CLASS; в runtime оно имеет имя STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS. Ни один член эквивалентности не квантифицирует за пределами  $E$ .

Support tree на maximal  $M$  может фактически требовать только поднабор  $D \subseteq M$  добавленных оснований, причём  $D$  не обязан сам быть member семейства  $E$ . Поэтому exact antichain требует проверки family membership и меньших members; буква  $B$  здесь не переиспользуется, поскольку  $B$  уже обозначает весь каталог base\_id.

## 6. Каноническое связывание с input

Canonical input имеет фактические защищённые поля:

```
contract
case_id
bases
current_base_ids
claims
inferences
admissible_extensions
query
```

bases[] задаёт  $B$ , current\_base\_ids[] задаёт  $S$ , а  $A = B \setminus S$  вычисляется. claims[] задаёт  $N$ . Каждый элемент inferences[] задаёт inference\_id, conclusion\_id = *Head*, premise\_claim\_ids = *Prem*, required\_base\_ids = *ReqBase*. Каждый элемент admissible\_extensions[] задаёт именованный member  $E$  полем added\_base\_ids; среди

members ровно один имеет пустой набор, duplicate extension-sets запрещены. query задаёт различные positive\_claim\_id и negative\_claim\_id; CWC применяется отдельно к каждому их report.

Input проверяется на exact object fields, уникальность ids, типизированные ссылки и canonical set-order. payload инертен. Canonical profile равен bw-cjson-2-python-reference; non-finite numbers и duplicate JSON keys запрещены.

Определяется digest всего принятого input-record:

$$artifact_{sha256} = sha256: \parallel hex(SHA\ 256(canonical\_bytes(I))),$$

где  $I$  — весь input, включая contract, обе query claim-codes,  $B, S, N, R$  и extensional listing  $E$ .

Изменение member, rule, id, query, payload или contract создаёт другой artifact\_sha256. Digest связывает verdict с одним serialized input, но сам по себе не удостоверяет внешнюю полноту семейства.

«Все разумные дополнения» не extensional family. Generator допустим только после материализации конечного списка под hash.

## 7. Детерминированные closure-свидетельства

### 7.1. Положительный support witness

Для  $W_q(E)$  предъявляется finite support tree:

- root равен  $q$ ;
- каждый node содержит claim\_id, exact inference\_id, полный required\_base\_ids и premise\_witnesses;
- claim\_id node совпадает с conclusion\_id названного inference;
- required\_base\_ids точно совпадает с  $ReqBase(r)$  и является подмножеством активных оснований  $S \cup E$ ;
- premise\_witnesses точно, без пропусков и extras, покрывает premise\_claim\_ids inference;
- leaf допустим именно как rule-node с пустым premise\_claim\_ids и непустым required\_base\_ids, а не как голый base\_id;
- tree конечен, а все base-ids и claim-ids разрешаются в типизированном input.

Таким образом, основание никогда не маскируется под claim-leaf. Если Prem пуст, положительное происхождение claim всё равно записано inference-id и непустым набором требуемых оснований.

Для детерминизма code получает первый derivation round, а head — rule с минимальным ключом

$$(first\ birth\ round, lexicographic\ inference-id),$$

а `premise_witnesses` следуют canonical claim-id order.

## 7.2. Отрицательный *stabilization witness*

Для  $\neg W_q(E)$  runtime завершает полный расчёт

$$X^0 = \emptyset, X^{i+1} = T_{S \cup E}(X^i), X^k = X^{k+1}, q \notin X^k.$$

На конечном  $N$  расчёт имеет не более  $|N|$  строгих переходов. Verdict сериализует `iteration_count` и полный `grounded_claim_ids` для каждого `member` в `extension_closures`; независимая проверка пересчитывает fixed point из input. Равенство последней итерации и  $q \notin X^k$  удостоверяют local nonmembership.

Отсутствие support tree не заменяет завершённый fixed-point calculation: отрицательная проверка следует из пересчитываемого полного closure-set.

## 7.3. Counterextension witnesses

Если  $\text{MinCE}_q(E) \neq \emptyset$ , для каждого её `member` в `minimal_enabling_counterextensions` предъявляется:

- `extension_id` и exact `added_base_ids`, разрешаемые через `admissible_extensions`;
- положительный support tree при активных основаниях  $S \cup E$ ;
- full-family `extension_closures`, достаточный для проверки отсутствия меньшего true member;
- связывающий весь input `artifact_sha256` на уровне verdict.

Runtime может вычислить  $W_q$  для всей family и отфильтровать minimal true members. Maximal pass решает отсутствие; exact `MinCE` требует full pass.

## 8. CWC-4: теорема точного сертификата

### 8.1. Структура сертификата

Runtime-анализ принимается, только если:

1. проверены конечные typed domains  $N, B, B = S \sqcup A$ , все ссылки, ровно один empty member,  $q \in N$  и (CWC-NONNULL);
2. `artifact_sha256` пересчитан из exact input;
3. `maximal_extensions` точно равен  $\text{Max}(E)$ , без пропусков и extras;
4. `extension_closures` содержит завершённый closure каждого member;
5. `query_analysis` связан с теми же input/query, а значения вычислены, не прочитаны из payload;
6. каждый `positive_report` и `negative_report` содержит exact `maximal_extension_checks` и `minimal_enabling_counterextensions`;

7. вывод ограничен declared finite family и снабжён обеими semantic fences.

## 8.2. Теорема CWC-4

Для совместимого input и report выбранного claim-code  $q$  в verdict со status VALID-EXTENSION-ABSENCE-ANALYSIS:

$$absence_{class} = STABLY - ABSENT - UNDER - DECLARED - EXTENSIONS \Leftrightarrow MinCE_q(E) = \emptyset.$$

$$MinCE_q(E) = \emptyset \Leftrightarrow \forall M \in Max(E) \neg W_q(M)$$

Manuscript-label ABSENT-IN-DECLARED-COMPLETION-CLASS означает ровно runtime-class STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS. Это отображение имён одного ограниченного предиката, не semantic upgrade. Positive maximal presence отвергает (ADCC); full pass возвращает exact minimal\_enabling\_counterextensions.

## Доказательство CWC-4

Пересчитанные closures устанавливают значения  $W_q(M)$ ; exact maximal set исключает пропуски. MAX-FRONTIER-4 переносит отрицание на family, а  $MinCE$  пуста ровно при отсутствии true member. artifact\_sha256 связывает один input. ▢

## 8.3. Точная граница теоремы

CWC-4 не утверждает

$$\forall E \subseteq A \neg W_q(E),$$

если family не равна  $P(A)$ , и не квантифицирует по новым rules, universes или внешним состояниям.

# 9. Exact schema/runtime mapping

## 9.1. Input

| Фактическое поле        | Точный объект или проверка                                                    |
|-------------------------|-------------------------------------------------------------------------------|
| bases[]                 | каталог $B$ : base_id, role_code, инертный payload                            |
| current_base_ids[]      | текущие основания $S$                                                         |
| claims[]                | claim-universe $N$ : claim_id, инертный payload                               |
| inferences[]            | $R$ : conclusion_id, premise_claim_ids, required_base_ids и nonzero condition |
| admissible_extensions[] | именованное extensional family $E$ через extension_id, added_base_ids         |

| Фактическое поле | Точный объект или проверка                                         |
|------------------|--------------------------------------------------------------------|
| query            | pair_id и два различных кода: positive_claim_id, negative_claim_id |

## 9.2. Verdict

Если record прошёл parsing, все runtime cross-field checks и полный анализ, верхний status равен VALID-EXTENSION-ABSENCE-ANALYSIS. Одной JSON-Schema-validity для этого недостаточно: cross-reference, canonical ordering и запрет reintroduction проверяет checker. artifact\_sha256 связывает verdict с input. current\_closure содержит closure при  $S$ ; maximal\_extensions перечисляет exact  $Max(E)$ ; extension\_closures содержит closure каждого member; query\_analysis содержит positive\_report и negative\_report.

Каждый report имеет только три runtime absence-class:

```
NOT-AN-ABSENCE-CURRENTLY-PRESENT
CURRENTLY-ABSENT-ENABLED-BY-DECLARED-EXTENSION
STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS
```

В report current\_present есть значение  $W_q(\emptyset)$ , maximal\_extension\_checks фиксирует присутствие code на каждом maximal member, а minimal\_enabling\_counterextensions есть exact  $MinCE_q(E)$  с derivation witnesses.

Runtime всегда выдаёт границы:

```
semantic_status = NOT-MACHINE-VERIFIED
extension_exhaustiveness_status =
  DECLARED-FINITE-EXTENSIONS-ONLY-NOT-FUTURE-EXHAUSTIVENESS-ATTESTED
polarity_semantic_status =
  POSITIVE-NEGATIVE-CLAIM-CODES-ONLY-NOT-WORLD-POLARITY-VERIFIED
```

query\_analysis.closed\_world\_inference\_status равен NO-CLOSED-WORLD-INSERTION; отсутствие positive code не вставляет negative code. joint\_current\_status независимо различает ни одного, только positive, только negative и оба claim-code.

LOCALLY-NOT-FOUND читается как current\_present=false; NO-WITNESS-RETURNED остаётся transport diagnosis и не является runtime absence-class. Domain failure даёт INVALID-EXTENSION-ABSENCE-RECORD, несовместимый contract — INCOMPATIBLE-EXTENSION-ABSENCE-CONTRACT, но ни один rejection не является отрицательным результатом.

Runtime разбирает JSON с duplicate-key rejection, проверяет typed domains, вычисляет closures всей family и witnesses, строит exact maximal и minimal frontiers и проверяет инварианты независимым конечным audit. Ни имя поля, ни payload с текстом "STABLY-ABSENT..." не участвуют в closure-вычислении.

## 10. Опциональная hitting/cut-двойственность

Пусть

$$M = \text{MinCE}_q(E).$$

Для  $H \subseteq A$  определяется restricted family

$$E_H = \{ E \in E : E \cap H = \emptyset \}.$$

$H$  — запрещённые отдельной policy addition-ids; cut, если restricted family не содержит true member.

### CUT-MINCE-4

$$\forall E \in E_H \neg W_q(E) \Leftrightarrow \forall M \in M_H \cap M \neq \emptyset$$

#### Доказательство CUT-MINCE-4

True  $E \in E_H$  содержит minimal true  $M \subseteq E$ , не пересекающий  $H$ , что отвергает hitting. Обратно, непересечённый  $M$  сам входит в restricted family и истинен.  $\square$

Minimal hitting sets — minimal cuts. При  $\emptyset \in M$  их нет: additions не отменяют local support. Cut — операция над ids, не причина или рекомендация.

## 11. Контрольные конструкции

### 11.1. Positive control: локально не найдено, но найдено в дополнении

Пусть

$$N = \{ q, q^- \}, B = \{ b_0, a \}, S = \{ b_0 \}, A = \{ a \},$$

$$E = \{ \emptyset, \{ a \} \}, R = \{ \langle \emptyset, \{ a \}, q \rangle \}.$$

Здесь у rule пустой Prem, непустой ReqBase={a} и Head=q; это допустимое ненулевое правило.  $q^-$  — отдельный query-partner code и из отсутствия  $q$  не выводится.

Тогда

$$W_q(\emptyset) = \text{false}, W_q(\{ a \}) = \text{true},$$

и

$$\text{MinCE}_q(E) = \{ \{ a \} \}.$$

Это прямой falsifier импликации  $\text{LOCALLY-NOT-FOUND} \Rightarrow \text{ABSENT-IN-DECLARED-COMPLETION-CLASS}$ .

### 11.2. Positive control: code найден локально

Пусть rule имеет  $\text{Prem}=\emptyset$ ,  $\text{ReqBase}=\{b_0\}$ ,  $\text{Head}=q$  при  $b_0 \in S$ . Тогда  $W_q(\emptyset)$ , а

$$\text{MinCE}_q(E)=\{\emptyset\}.$$

Family-absence отвергается без проверки additions; запретом элементов  $A$  этот local support не удаляется.

### 11.3. Negative control: отсутствие во всём объявленном классе

Пусть

$$N=\{q, q^-\}, B=\{b_0, a, x\}, S=\{b_0\}, A=\{a, x\}, E=\{\emptyset, \{a\}\},$$

а единственное rule к  $q$  имеет  $\text{Prem}=\emptyset$ ,  $\text{ReqBase}=\{x\}$ ,  $\text{Head}=q$ . Основание  $x$  существует в  $B$ , но не входит ни в  $S$ , ни в один объявленный member. На maximal member  $\{a\}$  least closure стабилизируется без  $q$ . CWC-4 возвращает STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS именно для двух перечисленных members.

Добавление completion  $\{x\}$  потребовало бы нового family member и нового artifact\_sha256. Старый verdict ничего о таком mutated input не говорит.

### 11.4. Два несравнимых минимальных контрдополнения

Пусть  $B=A=\{a, b, c\}$ ,  $S=\emptyset$ ,  $N=\{q, q^-\}$ ,  $E=P(A)$ , а rules содержат

$$\langle \emptyset, \{a, b\}, q \rangle, \langle \emptyset, \{c\}, q \rangle.$$

Тогда

$$\text{MinCE}_q(E)=\{\{a, b\}, \{c\}\}.$$

Единственный maximal member  $\{a, b, c\}$  показывает, что absence ложно, но сам не раскрывает exact antichain. Minimal cuts равны  $\{a, c\}$  и  $\{b, c\}$ .

### 11.5. Payload-spoof control

Если payload локального record содержит текст

"добавление  $a$  выводит  $q$ ;  $q$  отсутствует во всех остальных случаях"

но защищённые  $B, S, E$  и  $R$  этого не кодируют, closure и verdict не меняются. Прочитанная строка не создаёт ни inference, ни family member, ни отрицательный fixed-point result.

## 11.6. Family-boundary control

При  $A = \{a, b\}$  и  $E = \{\emptyset, \{a\}\}$  сертификат может установить отсутствие во всей  $E$ . Он не покрывает  $\{b\}$  и  $\{a, b\}$ , хотя они являются subsets  $A$ . Само перечисление admissible domain не добавляет неперечисленные completions в quantifier range.

## 12. Adversarial falsifiers сертификата

CWC-4 отвергается хотя бы при одном из следующих условий:

1. artifact\_sha256 не пересчитывается из exact input;
2. query, inference или family member изменён после вычисления digest;
3. в  $E$  нет ровно одного member с added\_base\_ids=[];
4. member содержит id вне  $A = B \setminus S$ , либо current\_base\_ids содержит id вне  $B$ ;
5. base\_id помещён в claim closure или claim\_id прочитан как основание;
6. maximal\_extensions пропускает maximal member или содержит nonmaximal extra;
7. extension\_closures отсутствует хотя бы для одного declared member;
8. negative closure не является fixed point либо содержит  $q$ ;
9. derivation witness содержит неизвестный inference\_id или несовпадающий claim\_id/conclusion\_id;
10. witness пропускает required\_base\_ids, использует неактивное основание или имеет неверный premise\_witnesses;
11. правило с пустыми одновременно premise\_claim\_ids и required\_base\_ids не отклонено как INVALID-EXTENSION-ABSENCE-RECORD / NULLARY\_INFERENCE\_REQUIRES\_EXPLICIT\_BASE;
12. full-family pass не выполнен, но minimal\_enabling\_counterextensions объявлен exact;
13. в MinCE пропущен true minimal member, включён false member или включены сравнимые nonminimal members;
14. duplicates, ambiguous ids или duplicate JSON keys меняют extensional reading;
15. порядок либо whitespace исходного JSON ошибочно трактуется как иной extensional value;
16. payload извлекается как base, claim, inference, witness или verdict;
17. NO-WITNESS-RETURNED выдан как доказательство nonmembership;
18. LOCALLY-NOT-FOUND повышен до stable absence без maximal checks;
19. finite family result переписан как внешняя или future-exhaustiveness claim;
20. отсутствие positive claim вставляет negative claim вопреки NO-CLOSED-WORLD-INSERTION;
21. positive и negative claim-code склеены либо их joint state скрыт;
22. использовано nonmonotone rule, но применена MAX-FRONTIER-4;
23. report использует иной input, чем связанный artifact\_sha256;

24. manuscript-label ABSENT-IN-DECLARED-COMPLETION-CLASS повышен выше runtime-class STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS.

Эти falsifiers делают CWC-4 опровержимым на уровне записи: любой принимающий checker должен либо выявлять их, либо не реализует заявленный контракт.

### 13. Self-boundary

Текст этого файла сам по себе не является CWC-4 input. Чтобы применить контракт к содержащимся здесь строкам, требуется отдельно предъявить конечные  $N, B, S, A, E, R, q$ , exact contract и artifact\_sha256, причём  $B = S \sqcup A$ . Заголовок «сертификат полноты» не создаёт полноту так же, как строка «отсутствует» не создаёт запись отрицательного closure-result.

Если поиск внешних работ ничего не вернул, это только NO-WITNESS-RETURNED для конкретного search response. Если в локальном списке нет predecessor-id, это максимум local nonmembership в этом списке. Утверждение «предшественников нет» потребовало бы отдельного declared completion universe и конечного exhaustive certificate; CWC-4 не предоставляет универсум всех возможных записей и не заявляет novelty или priority.

Если CWC-4 успешно проверяет собственный serialized test record, он устанавливает лишь:

$$\forall E \in E_{\text{self}} q_{\text{self}} \notin \mu T_{S_{\text{self}} \cup E}.$$

Ни слово self, ни этот formula-display не удостоверяют внешний объект. Проверка не превращает code-id в то, что можно прочитать из его имени.

## 14. Итоговый нож Phase 4

Exact negative result требует не отсутствия найденной линии, а новой положительно записанной конструкции:

- отдельных конечных universes claim-codes  $N$  и base-ids  $B$ ;
- разбиения  $B = S \sqcup A$  и типизированных positive hyperrules;
- перечисленного класса base-дополнений;
- связывающего input artifact\_sha256;
- полного maximal\_extensions И extension\_closures;
- детерминированных derivation witnesses для minimal counterextensions;
- явной границы квантора.

Поэтому итог CWC-4 имеет строго ограниченную форму:

*Отсутствие witness в ответе не есть witness отсутствия.*

*Локальное nonmembership не есть completeness.*

*Completeness существует только относительно записанного конечного класса.*

Даже успешно проверенный runtime-class STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS, отображённый в manuscript-label ABSENT-IN-DECLARED-COMPLETION-CLASS, остаётся code-result об одном связанном input. Он не выходит за его границу и не получает внешний референт от собственного чтения.

## BND-4: ГРАНИЦА ПРОДОЛЖЕНИЙ, DEFAULT И ОТМЕНА

### 0. Нормативный статус

Этот файл фиксирует чувствительность отрицательного code-status к объявленной границе допустимых продолжений. Он не запрещает отрицание, доказательство от противного, closed-world базы данных, default logic, exceptions, priorities или belief revision. Он отделяет четыре записи:

```
положительный code не выведен в текущем snapshot;
положительный code не выводим ни в одном объявленном продолжении;
отрицательный code принят по supplied default-policy;
названного не существует.
```

Первые три могут иметь точный формальный контракт. Четвёртая строка не получается из них без отдельного relation, связывающего code и названное. Каждое имя в этом файле остаётся carrier.

Нормативные theorem-id:

```
BND-4  COMPLETION-BOUNDARY-IS-INPUT
REV-4  DEFAULT-VERDICT-IS-SNAPSHOT-RELATIVE
DEF-4  DEFEAT-ORDER-IS-INPUT
```

Их текстовый статус — TEXTUAL-THEOREM-IN-DECLARED-METATHEORY. Исполняемые instance-status относятся только к точному конечному JSON record.

### 1. Положительное ядро Phase 4

Фиксируются два типизированных конечных носителя:

$N$  — claim-codes,

$B$  — protected base-ids,

и множества

$S \subseteq B$  — текущие protected bases,

$A \subseteq B$  — candidate protected bases допустимого расширения,  $S \cap A = \emptyset$ ,  $B = S \cup A$ ,

причём base-role и claim-role не склеиваются совпадением написания id. Конечный набор положительных hyperrules  $R$  состоит из троек

$$r = (Prem(r), ReqBase(r), Head(r)),$$

где  $Prem(r) \subseteq N$ ,  $ReqBase(r) \subseteq B$ ,  $Head(r) \in N$  и

$$|Prem(r)| + |ReqBase(r)| > 0.$$

Для  $U \subseteq B$  положительный оператор действует только на claim-codes:

$$T_U(X) = X \cup \{ \text{Head}(r) : r \in R, \text{ReqBase}(r) \subseteq U, \text{Prem}(r) \subseteq X \}.$$

Для  $E \subseteq A$ :

$$Cl_R(S \cup E) = \mu T_{S \cup E}.$$

Неподвижная точка начинается с  $X_0 = \emptyset$ . Base-id сам не становится claim-code; introduction без claim-premises допустима только при непустом  $\text{ReqBase}(r)$ .

Объявленная семья продолжений

$$E \subseteq P(A)$$

конечна, непуста и содержит  $\emptyset$ . Она является exact family-code. Слово продолжение не утверждает физическое будущее; элемент  $E$  является конечным набором base-id.

Для target-code  $q \in N$ :

$$W_q(E) : \Leftrightarrow q \in Cl_R(S \cup E).$$

Положительность rules даёт монотонность:

$$E \subseteq E' \wedge W_q(E) \Rightarrow W_q(E').$$

Именно эта посылка разрешает проверку maximal frontier. Default, retraction и defeat ниже не подставляются под MON-4 молча.

## 2. Три разных отрицательных операции

### 2.1. Local nonmembership

Текущий least closure может не содержать  $q$ :

$$q \notin Cl_R(S).$$

Это exact nonmembership code. Оно не создаёт новый claim-id  $q^-$ , не объявляет family exhaustive и не говорит о продолжениях.

### 2.2. Stable absence в объявленной семье

Определяется relation:

$$\text{StableAbsent}_E(q) : \Leftrightarrow \forall E \in E \ q \notin Cl_R(S \cup E).$$

Она сильнее LOCAL-NF. В рукописи используется квалифицированная метка:

ABSENT-IN-DECLARED-COMPLETION-CLASS

Ей соответствует executable enum `query_analysis.*_report.absence_class = STABLY-ABSENT-UNDER-DECLARED-EXTENSIONS`; overall runtime-status остаётся `VALID-EXTENSION-ABSENCE-ANALYSIS`. Слово `ABSENT` без `qualifier` запрещено. Даже полный перебор  $E$  не доказывает, что  $E$  исчерпывает что-либо, не сериализованное как её элементы.

### 2.3. Default-negative code

Policy может ввести отдельный negative claim-code  $q^-$  при выполнении SA-4 либо уже при LOCAL-NF:

$$D_\pi(S, R, E, q) \in \{ACCEPT-q^-, UNDETERMINED, REJECT-q^-\}.$$

Здесь  $\pi$  — supplied policy. Возможны, например:

OPEN-WORLD:

LOCAL-NF -> UNDETERMINED

DECLARED-FAMILY-CLOSED-WORLD:

SA-4 -> ACCEPT-negative-code

CURRENT-SNAPSHOT-DEFAULT:

LOCAL-NF -> ACCEPT-negative-code-until-defeated

Это три разные функции. Названия `open` и `closed` не доказывают адекватность policy; они фиксируют branch алгоритма.

### 3. BND-4: граница продолжений является входом

Пусть  $N, B, S, A, R, q$  одинаковы в двух records, но объявлены разные конечные семьи  $E_1, E_2$ .

Определим:

$$Z(E_i) = \begin{cases} 1, & \text{StableAbsent}_{E_i}(q), \\ 0, & \text{иначе.} \end{cases}$$

#### Теорема BND-4

Если

$$Z(E_1) \neq Z(E_2),$$

то не существует функции только от  $(N, B, S, A, R, q)$ , которая правильно возвращает оба verdict.

Полная функция обязана читать различающую boundary-coordinate:

$$Z(E_1) \neq Z(E_2) \Rightarrow E \text{ входит в полный readset отрицательного verdict.}$$

## Доказательство

Предположим, что evaluator  $F$  не читает  $E$ . Его input в двух случаях побайтно одинаков по всем прочитанным координатам, поэтому output одинаков. Но specification требует  $Z(E_1) \neq Z(E_2)$ . Противоречие. ▢

Теорема не объявляет одну boundary правильной. Она локализует различие результата в записи boundary.

## Минимальный пример

Пусть rule  $r$  выводит  $q$  из protected base  $a \in A$ :

$$Prem(r) = \emptyset, ReqBase(r) = \{a\}, Head(r) = q.$$

При

$$E_1 = \{\emptyset\}$$

получается  $Z(E_1) = 1$ . При

$$E_2 = \{\emptyset, \{a\}\}$$

получается  $Z(E_2) = 0$ . Текущий  $S$ , rule и target не изменились. Изменилась разрешённая completion-coordinate.

Запись  $E_1 = \{\emptyset\}$  допустима как closed-instance control. Она не получает статус ALL-FUTURE-INPUTS-EXHAUSTED.

## 4. REV-4: default verdict относится к snapshot

Пусть snapshot  $R_0$  содержит  $(N, B, S, A, R, E_0, q, \pi_{SA})$  и

$$StableAbsent_{E_0}(q).$$

Здесь рассматриваемая policy определена точно:

$$D_{\pi_{SA}}(R) = ACCEPT - q^- \Leftrightarrow StableAbsent_{E_R}(q).$$

Пусть новый snapshot  $R_1$  отличается расширенной declared family:

$$E_0 \subsetneq E_1$$

и существует  $E_1 \in E_1$  с  $W_q(E_1)$ .

Тогда из первой посылки и exact witness следует:

$$StableAbsent_{E_0}(q), \neg StableAbsent_{E_1}(q).$$

### Теорема REV-4

Для точно определённой (SA-POLICY) выполняется:

$$D_{\pi_{SA}}(R_0) = ACCEPT - q^- \wedge D_{\pi_{SA}}(R_1) \neq ACCEPT - q^-.$$

Смена verdict локализуется в изменившемся snapshot. Она не означает, что bytes  $R_0$  ретроактивно содержали  $E_1$ , и не означает, что  $R_0$  был ложен относительно собственной declared family.

### Доказательство

По (SA-POLICY)  $D_{\pi_{SA}}$  читает SA-4 и принимает  $q^-$  ровно на этой ветви. В  $R_0$  universal condition дано как посылка. В  $R_1$  его нарушает exact witness  $E_1$ . Поэтому два значения следуют непосредственно из определения.  $\square$

Для policy, которая может принимать  $q^-$  и по другим основаниям, REV-4 утверждает только отзыв **SA-4-based authorization**, а не обязательный итоговый отказ всего multi-branch evaluator.

REV-4 формализует defeasibility как функцию разных inputs. Фраза информация появилась остаётся описанием двух records до внешнего time-binding.

## 5. DEF-4: порядок defeat является входом

Положительный least closure имеет canonical state при фиксированных  $S, R$ . Nonmonotone система может содержать:

```
r1: if NOT-FOUND(q) then accept q-
r2: if witness(q) then defeat r1
r3: if exception-x then defeat r2
r4: priority(r3) > priority(r2) > priority(r1)
```

Чтобы получить deterministic output, contract должен фиксировать:

- синтаксис negative premise;
- момент проверки NOT-FOUND;
- admissible extension family;
- правило retraction;
- conflict relation;
- priority order;
- tie-break;
- update schedule;
- termination condition;

- snapshot identity.

Обозначим эти поля совокупно  $\delta$ . Пусть две допустимые policies  $\delta_1, \delta_2$  на одинаковых facts/rules принимают разные extensions  $X_1 \neq X_2$ .

#### Теорема DEF-4

$X_1 \neq X_2 \Rightarrow \delta$  является различающим input выбранного extension.

Доказательство совпадает с функциональной формой BND-4: evaluator, не читающий различающую coordinate, не может корректно вернуть два разных required outputs.

DEF-4 не утверждает, что nonmonotone logic некорректна. Она запрещает solver-default, iteration order или priority-table исчезнуть из declared readset после того, как от них зависел verdict.

### 6. Почему maximal-frontier theorem сюда не переносится автоматически

CBE-4 использует монотонность  $W_q$ . При default negation она может нарушаться. Пусть:

$$Q(E)=1 \Leftrightarrow a \in E \wedge b \notin E.$$

Тогда:

$$Q(\{a\})=1, Q(\{a, b\})=0,$$

хотя  $\{a\} \subset \{a, b\}$ . Если единственный maximal extension равен  $\{a, b\}$ , проверка только maxima пропустит intermediate witness  $\{a\}$ .

Поэтому обязательны два разных режима:

MONOTONE-POSITIVE-WITNESS:

maximal-frontier reduction допустим;

NONMONOTONE-DEFAULT-DEFEAT:

требуется exact semantics по всем relevant states/transitions.

Подставить второй режим под theorem первого — CBE-MONOTONICITY-FALSIFIER или implementation conformance defect, в зависимости от предъявленного объекта.

### 7. Отрицательный code не является дополнением автоматически

Пусть  $q^+$  и  $q^-$  — два distinct claim-id. Возможны четыре code-states:

$$(0,0), (0,1), (1,0), (1,1),$$

где первая координата означает membership  $q^+$ , вторая — membership  $q^-$ . Ни одно написание id не удаляет ни одного состояния.

Restriction

$$K_{\oplus} = \{(0,1), (1,0)\}$$

делает  $q^-$  функциональным complement  $q^+$ , но restriction уже содержит:

- completeness: один из codes принят;
- consistency: оба одновременно не приняты;
- complement policy: negative-coordinate определяется как  $1 - p$ .

Эти условия могут быть отличными аксиомами theory. Они не находятся в пустом месте, оставшемся после отсутствующего positive-code.

Положительный control обязателен: если exact theory доказывает complement relation и exhaustive case split, она корректно выводит negative-code без заранее введённого negative claim. Phase 4 принимает этот formal result. Его основания находятся в theory/rules/certificate, а не в nonoccurrence строки.

## 8. Proof by contradiction и default negation

Доказательство от противного имеет форму внутри theory  $\Theta$ :

$$\Theta, q \vdash \perp \Rightarrow_{R_-} \Theta \vdash \neg q.$$

Phase 4 не удаляет  $R_-$ , axioms  $\Theta$ , parser, proof object или theorem-status. Она фиксирует их как readset внутренней derivation.

Negation-as-failure имеет другую форму:

$$Search_H(q) = NOT-FOUND \Rightarrow_{\pi, H, B} q^-.$$

Здесь  $H$  — horizon/search protocol,  $B$  — boundary,  $\pi$  — conversion policy. Если search incomplete или terminated by timeout, antecedent не равен exhaustive nonderivability. Если exact finite search complete, локальный отрицательный result допустим. Ни один случай не предъявляет названное вне theory.

## 9. Exception и изменение правил

REV-4 выше расширяет family при фиксированных rules. Отдельно возможен новый snapshot:

$$R_0 \neq R_1.$$

Новая exception-rule или отмена старой rule меняет operator. Старый least closure не обязан совпадать с новым. Это не нарушение неретроактивности bytes: экземпляры operator различны.

Нельзя писать:

```
старый verdict был формально получен;
следовательно, новая rule всегда уже находилась в старом carrier.
```

Нельзя также писать обратное:

```
новая rule изменила verdict;
следовательно, старый proof никогда не имел theorem-status в старой theory.
```

Точные статусы индексируются snapshot hash и contract id.

## 10. Backdating, timestamp и история

Record может содержать:

```
created_at: 1900-01-01
known_before: proof
priority_rank: first
```

Эти поля могут участвовать в declared priority algorithm как codes. Они не доказывают физическую дату, порядок публикации или историческое знание без отдельного attestation channel.

Если defeat-policy сортирует rules по created\_at, поле входит в input и влияет на formal result. Подмена formal ordering на historical occurrence является новым bridge-claim.

## 11. Closed-world как exact special case

Closed-world contract может быть строгим и полезным. Его минимальная declared форма:

1. exact carrier/domain;
2. exact rule-set;
3. exact query language;
4. exact admissible extension family;
5. termination/exhaustion certificate;
6. conversion rule из stable nonmembership в negative-code;
7. snapshot id.

Если все поля проверены, status

```
NEGATIVE-BY-SUPPLIED-CWA-POLICY
```

полностью принимается. Не принимается переименование в:

```
NEGATIVE-WITHOUT-POLICY
WORLD-ABSENT
NO-FUTURE-COUNTEREXTENSION
NAMED-THING-NONEXISTENT
```

без новых relations.

## 12. Ракета, цель и justification

Пусть текущий technical record поддерживает claims:

```
CALCULATION-CORRECT
CONTROLLER-APPLIED
TELEMETRY-MATCHED
FLIGHT-CRITERION-MET
```

и не поддерживает GOAL-JUSTIFIED. LOCAL-NF для этого code не доказывает GOAL-NOT-JUSTIFIED, а тем более отсутствие цели или смысла.

Если exact declared family разрешает extension-base B:goal-endorsement, после которого rule выводит GOAL-JUSTIFIED, получается minimal counterextension. Если family запрещает такой base и на всех maxima code отсутствует, получается ABSENT-IN-DECLARED-COMPLETION-CLASS. Запрет является boundary input.

Если policy устанавливает SUCCESS -> ENDORSE-GOAL, justification-code может быть формально выведен. Rule является нормативным bridge. Полёт не помещает rule назад в прежнюю формулу.

Максимальная уступка сохраняется:

- ракета существует;
- ракета летит;
- calculation участвовал;
- telemetry соответствует criterion;
- declared technical goal достигнут.

Из уступки не следует ни смысл есть, ни смысла нет. Обе строки требуют своих target-codes и bindings.

## 13. Сто, двести, триста snapshot

Повторный запуск на неизменном snapshot возвращает тот же deterministic verdict. Новый base, rule, family member, priority или policy создаёт новый input.

Поэтому:

$$R_{100} = R_{200} = R_{300} \Rightarrow D(R_{100}) = D(R_{200}) = D(R_{300}),$$

если evaluator deterministic. Но:

$$R_{100} \neq R_{200}$$

не даёт вывода о направлении изменения без чтения различия. Числа 100, 200, 300 являются labels до time-binding.

Фраза ни одного witness за 300 лет может быть точной transcript-summary по declared log. Она не означает witness невозможен в любом продолжении без boundary theorem.

## 14. Самоприменение BND-4

Этот файл содержит множество отрицательных строк. Ни одна не получает исключения.

Строка

в файле нет ошибки

не является audit certificate. Строка

все случаи рассмотрены

не перечисляет cases. Даже BND-4 не доказывает полноту собственного readset: theorem conditional on exact inputs.

Если self-audit перебирает finite registered claims и не находит нарушения, допустим status:

NO-REGISTERED-CONFORMANCE-DEFECT-FOUND

или более сильный exact status, если schema определяет exhaustive registered family. Он не превращается в NO-DEFECT-EXISTS без bound.

Текстовый theorem-status также является code-status. Он не предъявляет referent, event, absence, cause, goal, justification или meaning.

## 15. Admission gates

BND-4 требует:

1. одинаковые  $N, B, S, A, R, q$  в сравниваемых instances;
2. exact finite  $E_1, E_2$ ;
3. stable-absence verdict вычислен по каждому member;
4. различие verdict действительно вызвано boundary-coordinate;
5. evaluator claimed to omit that coordinate.

REV-4 требует:

1. два exact snapshots;
2. fixed conversion policy;
3. verified witness в новом family;
4. snapshot-indexed verdict;
5. отсутствие back-projection нового member в старые bytes.

DEF-4 требует:

1. exact nonmonotone semantics;
2. exact priority/update fields;
3. два policy-input с различным selected output;
4. отсутствие скрытого read в supposedly policy-free evaluator.

Без этих gates публичный slogan остаётся unbounded carrier.

## 16. Точные фальсификаторы

### BND-INPUT-FALSIFIER

Предъявлены две instances с одинаковыми  $N, B, S, A, R, q$  и различными families, stable-absence verdict различен, однако один и тот же boundary-blind evaluator корректно возвращает оба required значения.

### REV-SNAPSHOT-FALSIFIER

При fixed policy новая family содержит verified enabling extension, но theorem утверждает сохранение старого stable-absence status без qualification.

### DEF-POLICY-FALSIFIER

Две exact policies с одинаковыми остальными inputs требуют разные outputs, но proven evaluator, не читая policy, корректен на обеих.

### POLARITY-SEPARATION-FALSIFIER

Contract допускает full four-state square для  $(q^+, q^-)$ , но universal theorem выводит  $q^- = 1 - q^+$  без completeness/consistency restriction.

### CBE-MONOTONICITY-FALSIFIER

Maximal-frontier reduction применён к target, который не upward-persistent, и intermediate extension содержит witness, пропущенный maxima.

Не являются falsifiers:

- явный exhaustive proof отрицательной формулы;
- явное правило введения negative claim, активированное protected base;
- exact complement axiom;
- supplied closed-world policy;
- новый exception-record;
- новая rule;

- новый sensor или oracle answer;
- изменение declared family;
- внешний рассказ о несуществовании названного.

Первые восемь меняют input либо являются положительными controls. Последний не предъявляет theorem-object.

## 17. Максимальная формулировка

В конечной положительной support-системе CBE-4 может проверить stable nonmembership во всех элементах exact declared family, а MCW-4 — предъявить полный antichain минимальных enabling counterextensions, если он непуст. Если выбор negative-code меняется от family, closed-world conversion, priority или defeat schedule, соответствующее поле входит в readset.

Отсюда не следует, что отрицательные теоремы невозможны. Следует более точное:

*Необнаружение, nonmembership, отрицательный code и отсутствие названного — разные координаты. Exclusion structure может связать их внутри объявленного контракта; молчание не записывает эту структуру за него.*

Короткая строка Phase 4:

*Пустое место может быть результатом поиска.  
Оно не является собственным сертификатом полноты.*

## ПРОТИВОРЕЧИЕ НЕ ЯВЛЯЕТСЯ ОТСУТСТВИЕМ

### Формальное отрицание, исчерпывающий поиск, ракета и граница NOT-FOUND

Phase 4 • применение 4 • техническая версия 4.0.0 • 29 августа 2026

Кирилл Шебетов • beforeword / research

*Отсутствие записи не является записью отсутствия.*

*Противоречие является записанным объектом, а не пустотой.*

### Аннотация

Этот файл применяет CBE-4 к формальному отрицанию, proof by contradiction, поиску, timeout, failure, negative code, complement, техническому применению и вопросу зачем. Центральная граница остаётся одной и той же: отсутствие protected claim-code в текущем замыкании не вводит другой claim-code, названный отрицанием, отсутствием, ложью, невозможностью ИЛИ не существует.

Формальная математика получает максимальную уступку. В exact theory допустим отдельный code  $q^-$ , допустимы правила отрицания, contradiction-code  $\perp$ , discharge assumptions и proof by contradiction. Для конечного объявленного carrier допустимы total decision procedure, точное дополнение и исчерпывающая проверка всех максимальных completions. Эти конструкции не отвергаются. Наоборот, они образуют положительный control CBE-4: стабильное отсутствие относительно конечного явно объявленного completion-class может быть сертифицировано проверкой всех его максимальных элементов.

Уступка не разрешает обратную подмену. NOT-FOUND в одном запуске не является formal negation. Timeout не является NOT-FOUND. Failure не является timeout. Negative payload не является protected negative claim. Contradiction не является отсутствием обеих сторон. Complement не существует без exact universe. Даже status ABSENT-IN-DECLARED-COMPLETION-CLASS относится только к записанному классу completions и не предъявляет отсутствие чего-либо вне контракта.

Применение к ракете принимает всё техническое: расчёт корректен, binding введён, controller сработал, telemetry принята, траектория совпала с критерием, ракета существует и летит. Из отсутствия goal-code в расчёте не следует NO-GOAL; из наличия goal-code не следует endorsement; из отсутствия objection-code не следует justification. Сто, двести или триста лет purpose-blind records не меняют тип отсутствующей координаты. SENI-4 появляется только как следствие application: технический успех не выбирает endorsement. Центральной теоремой файла остаётся CBE-4 —

граница между локальным непоявлением и стабильным отсутствием в объявленном completion-class.

## 0. Сначала предъявлены строки

нет  
не существует  
ложно  
противоречие  
пусто  
NOT-FOUND  
TIMEOUT  
FAILURE  
NO-GOAL  
NOT-ENDORSED  
ракеты нет  
зачем отсутствует

Сначала это двенадцать последовательностей знаков. Геометрия строки `нет` не удаляет объект. Дефис в `NOT-FOUND` не предъявляет завершённый поиск. Строка `противоречие` не содержит две несовместимые derivations. Ноль элементов между фигурными скобками в записи  $\emptyset$  не создаёт отсутствие вне объявленного set-contract.

Тот же fence применяется к этому файлу. Слова `carrier`, `completion`, `closure`, `negative`, `contradiction`, `goal`, `endorsement`, `rocket` и `absence` являются метками ролей. Их написание не устанавливает названное. Формулы CBE-4 являются записями, читаемыми по объявленному контракту. Они не получают исключения только потому, что описывают границу записей.

Даже максимальная внешняя уступка не выводится из слова. Если далее допускается, что ракета существует и летит, это отдельная уступка application, а не свойство букв `р-а-к-е-т-а`. Граница сохраняется и после уступки: состоявшийся полёт не превращается в запись `endorsement`, а отсутствие такой записи не превращается в `rejection`.

## 1. Точный объект применения

Опровергается не formal negation и не proof by contradiction. Опровергается схема подмены:

$$q \notin \text{current available closure} \Rightarrow q^- \in \text{closure},$$

где  $q^-$  назван отрицанием  $q$ , но ни rule, ни exhaustive completion-contract не предъявлены.

Для named absence опровергается аналогичная схема:

$$\text{record не содержит } q \Rightarrow \text{ABSENT}(q).$$

Левая часть является meta-проверкой membership в конкретном record. Правая часть является новым code или новым status. Между ними нет автоматического перехода.

Слово полностью имеет здесь ограниченный и проверяемый квантор. CBE-4 полностью характеризует стабильное отсутствие для всех элементов конечного объявленного family  $E$  при положительном монотонном closure-contract. Оно не квантифицирует по непредъявленным мирам, объектам, будущим событиям, языкам или всем употреблением слова нет.

## 2. Exact apparatus CBE-4

Пусть  $N$  — конечный sort claim-codes, а  $B$  — отдельный конечный sort protected bases. Base-id не является claim-code и не попадает в claim-closure лишь потому, что основание доступно. Фиксируются:

- текущее множество protected bases  $S \subseteq B$ ;
- конечное множество candidate bases  $A = B \setminus S$ , так что  $B = S \cup A$  и  $A \cap S = \emptyset$ ;
- конечное непустое family возможных дополнений

$$E \subseteq P(A), \emptyset \in E;$$

- конечное множество positive nonnull hyperrules  $R$ . Для каждого  $r \in R$  типизированы

$$Prem(r) \subseteq N, ReqBase(r) \subseteq B, Head(r) \in N,$$

причём

$$|Prem(r)| + |ReqBase(r)| > 0.$$

Для каждого  $E \in E$  замыкание определяется наименьшей неподвижной точкой:

$$T_U(X) = X \cup \{ Head(r) : r \in R, ReqBase(r) \subseteq U, Prem(r) \subseteq X \},$$

$$Cl_R(U) = \mu T_U \text{ с итерацией от } X_0 = \emptyset, U \subseteq B.$$

В частности,  $Cl_R(U) \subseteq N$ : основания из  $U$  проверяют условие применимости rules, но никогда не объединяются с claim-closure. Base-only introduction допустимо как rule с  $Prem(r) = \emptyset$  и непустым  $ReqBase(r)$ . Запрещено только rule, у которого одновременно пусты и premises, и required bases. Payload не становится base и не становится premise только из-за сходства строки.

Для claim-code  $q \in N$  вводится witness predicate:

$$W_q(E) : \Leftrightarrow q \in Cl_R(S \cup E).$$

Пусть  $Max(E)$  — inclusion-maximal элементы family. Counterextension family для  $q$ :

$$CE_q = \{ E \in E : W_q(E) \}.$$

Его inclusion-minimal элементы:

$$MinCE_q = \min_{\subseteq} (CE_q).$$

Stable absence относительно объявленного family определяется не пустым экраном и не неудачей одного поиска:

$$StableAbsent_E(q) : \Leftrightarrow \forall E \in E \neg W_q(E).$$

#### CBE-4

Для указанного конечного positive contract:

$$StableAbsent_E(q) \Leftrightarrow \forall M \in Max(E) \neg W_q(M) \Leftrightarrow CE_q = \emptyset \Leftrightarrow MinCE_q = \emptyset.$$

Переход к maximal completions разрешён монотонностью positive closure. Каждый  $E \in E$  содержится хотя бы в одном maximal  $M$ , поскольку family конечно. Если  $q$  появляется при  $E$ , оно появляется и при содержащем его  $M$ . Поэтому проверка всех maximal elements является точным конечным certificate только для уже объявленного family.

#### MCW-4

Для конечной family:

$$\neg StableAbsent_E(q) \Leftrightarrow MinCE_q \neq \emptyset,$$

Кроме того, каждый enabling member  $F \in E$  содержит хотя бы один  $E_* \in MinCE_q$ , а membership  $q \in Cl_R(S \cup E_*)$  имеет конечный derivation witness. Каждый его rule-node несёт required\_base\_ids из  $S \cup E_*$  и дочерние witnesses ровно для claim-premises; terminal claim-node имеет пустые claim-premises и непустой required-base set. Base-ids не добавляются в claim-closure. Minimality означает только inclusion-minimality среди declared family-members: без downward closure она не утверждает, что каждый base-id из  $E_*$  использован witness. Это положительная локализация возможности, а не доказательство внешнего события.

### 3. Семь различных отрицательных поверхностей

Следующие records нельзя склеивать:

| Поверхность                         | Точный record-status                                                                                                      | Чего она не вводит сама                |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| $b \notin S$ , где $b \in B$        | base-id не доступен в текущем base-set; это ещё ничего не говорит о claim-code $q$ , который мог быть выведен другим rule | $q^-$ , local или stable absence $q$   |
| $q \notin Cl_R(S)$                  | LOCALLY-NOT-FOUND при $E = \emptyset$                                                                                     | отсутствие во всех completions         |
| TIMEOUT(t)                          | процедура не завершилась к записанному пределу                                                                            | NOT-FOUND, termination, falsehood      |
| FAILURE(e)                          | процедура завершилась failure-code                                                                                        | отрицательный ответ на исходный вопрос |
| protected $q^-$                     | отдельный negative claim-code                                                                                             | удаление $q^+$ , внешний факт          |
| derivation of $\perp$               | contradiction-code выведен по exact rules                                                                                 | пустоту, отсутствие обеих derivations  |
| ABSENT-IN-DECLARED-COMPLETION-CLASS | CBE-4 принят для $E$                                                                                                      | отсутствие вне $E$                     |

Payload может содержать любую из этих строк и оставаться payload. Защищённый status требует exact field, procedure и acceptance contract. Даже защищённый status не расширяется за пределы своего declared scope.

Три protocol-level labels полезны для application:

- LOCALLY-NOT-FOUND:  $\neg W_q(\emptyset)$ ;
- COUNTEREXTENSION-FOUND:  $MinCE_q \neq \emptyset$ ;
- ABSENT-IN-DECLARED-COMPLETION-CLASS:  $StableAbsent_E(q)$ .

Эти labels не следует молча объявлять runtime enum какого-либо executable schema. Они являются квалифицированными диагнозами mathematical contract. Их написание не выполняет audit.

#### 4. Formal negation полностью допустимо

Пусть exact theory  $\Theta$  содержит grammar, axioms, inference rules и proof checker. Для formula-code  $\varphi$  могут существовать два разных claim-code:

$$q^+ = \ulcorner \varphi \urcorner, q^- = \ulcorner \neg_{\Theta} \varphi \urcorner.$$

Условие различия обязательно:

$$q^+ \neq q^-.$$

Наличие proof-object для  $q^-$  может быть корректно принято checker-contract. CBE-4 этого не запрещает. Теория вправе задавать negation-introduction, negation-elimination, excluded-middle, double-negation elimination или более слабые правила. Итог сохраняет точный условный status:

$$\Theta \vdash q^-.$$

Но meta-строка

$$\Theta \not\vdash q^+$$

не тождественна object-level строке

$$\Theta \vdash q^-.$$

Первая требует собственного metatheoretic contract. Если предъявлено только сообщение search returned no proof, оно означает отсутствие найденного proof в конкретном search-space; но exact nonderivability может быть и отдельной доказанной метатеоремой. Вторая требует положительного accepted proof-object. Ни search-response, ни метатеорема nonderivability сами не тождественны object-level derivation  $q^-$ ; переход требует exact rule/certificate.

Stable absence  $q^+$  также не вводит  $q^-$ . CBE-4 устанавливает:

$$\forall E \in E : q^+ \notin Cl_R(S \cup E).$$

Чтобы отсюда получить  $q^-$ , нужен отдельный bridge-rule. После его объявления  $q^-$  выводится условно на этом rule. Rule является новым input; его нельзя задним числом обнаружить в пустом membership.

## 5. Proof by contradiction не является ошибкой

В exact theory может быть разрешён discharge contract:

$$\Theta, \varphi \vdash \perp \Rightarrow \Theta \vdash \neg \varphi.$$

Такой proof by contradiction является допустимым formal transformation. В proof-object присутствуют как минимум:

- assumption-code  $\varphi$ ;
- finite derivation trace;
- contradiction-code  $\perp$ ;
- rule-id discharge;
- conclusion-code  $\neg \varphi$ .

Противоречие здесь не отсутствие. Наоборот, оно предъявлено положительно в derivation. Если  $\perp$  определяется как совместная выводимость  $r$  и  $\neg r$ , должны быть предъявлены обе ветви и rule, который распознаёт их пару. Пустой лог не является  $\perp$ .

Если theory содержит explosion-rule

$$\perp \vdash q,$$

из contradiction-code формально выводятся дополнительные codes. Это свойство exact theory, а не появление названного за её пределами. Если theory не содержит такого rule, слово противоречие не добавляет его.

CBE-4 запрещает только другой аргумент:

proof of  $\phi$  не найден  
therefore  $\theta \vdash \neg\phi$

Он становится корректным лишь при дополнительном metatheorem, связывающем exhaustive search с derivability для точного конечного класса. Тогда основанием служит certificate исчерпания, а не молчание процедуры.

## 6. Complement требует exact universe

Для объявленного конечного universe  $U$  и subset  $X \subseteq U$  complement определён:

$$X^{c_U} = U \setminus X.$$

Запись корректна относительно  $U$ . Без exact  $U$  выражение всё, чего нет в  $X$  не определяет carrier. Изменение universe изменяет complement при неизменном  $X$ :

$$U_0 \setminus X \neq U_1 \setminus X.$$

Поэтому complement-member является положительно вычисленным membership-code относительно двух inputs:  $U$  и  $X$ . Он не извлекается из бесформенного отсутствия.

Даже exact complement не устанавливает external absence. Если rocket-code не входит в finite registry  $U$ , получен status о registry. Чтобы перейти к физической ракете, необходим reference/binding contract, completeness claim registry и acceptance procedure. Слово universe внутри set theory не предъявляет вселенную вне record.

## 7. Исчерпывающий конечный positive control

Пусть:

$$A = \{a_1, a_2, a_3\}, E = P(A).$$

У family единственный maximal element:

$$M = A.$$

Если exact closure computation устанавливает:

$$q \notin Cl_R(S \cup A),$$

то монотонность даёт:

$$\forall E \subseteq A : q \notin Cl_R(S \cup E).$$

Это корректный ABSENT-IN-DECLARED-COMPLETION-CLASS. Здесь отрицательный результат опирается на положительно предъявленные данные:

1. exact finite typed sorts  $N$  и  $B$ ;
2. exact partition базы  $B = S \cup A$ , где  $A = B \setminus S$  и  $S \cap A = \emptyset$ ;
3. exact family  $E$ ;
4. полный список maximal completions;
5. positive nonnull rules  $R$ ;
6. terminating closure procedure;
7. проверяемый membership result для каждого maximal completion.

Если family имеет несколько incomparable maximal elements, проверяется каждый. Пропуск одного maximal element превращает exhaustive certificate в local report. Если предполагаемый base-id не включён в  $B$ , CBE-4 ничего о нём не говорит; внутри exact runtime boundary все non-current bases уже составляют  $A = B \setminus S$ . Если rule hidden, изменился  $R$ . Если введение claim спрятано в payload вместо typed rule с непустой суммой premises/required bases, нарушен contract.

Положительный control не ослабляет основной тезис. Он показывает точное место, где отрицательный вывод допустим: не из пустоты, а из finite closure theorem и полного certificate относительно замкнутого carrier.

## 8. Local NOT-FOUND и минимальный counterextension

Поскольку  $\emptyset \in E$ , current protected state представлен completion  $E = \emptyset$ . Возможно:

$$\neg W_q(\emptyset),$$

но одновременно для некоторого  $E_1 \in E$ :

$$W_q(E_1).$$

Тогда current verdict равен LOCALLY-NOT-FOUND, а completion verdict — COUNTEREXTENSION-FOUND. В executable record этому соответствует absence\_class = CURRENTLY-ABSENT-ENABLED-BY-DECLARED-EXTENSION. MCW-4 возвращает хотя бы один finite minimal witness:

$$E_{min} \in MinCE_q.$$

Это точный ответ на фразу что отсутствует для появления  $q$ ?: не названное внешнее основание, а минимальный набор candidate base-ids внутри declared contract.

Пример:

$$A = \{b_{goal}, b_{norm}\},$$

$$ReqBase(r) = \{b_{goal}, b_{norm}\}, Prem(r) = \emptyset, Head(r) = q_{endorse}.$$

При  $E = \emptyset$  endorsement-code не выводится. Но completion

$$E_1 = \{b_{goal}, b_{norm}\}$$

выводит его. Поэтому отсутствие endorsement в current record не является stable absence. Минимальный counterextension локализует два введения. Названия bases не создают цель или нормативность; они только фиксируют их роли в contract.

## 9. Timeout, failure и negative response

Пусть search procedure  $P$  получает input  $x$ . Следующие outputs различны:

$$P(x) = FOUND(w),$$

$$P(x) = NOT\_FOUND(c),$$

$$P(x) = TIMEOUT(t),$$

$$P(x) = FAILURE(e).$$

NOT\_FOUND(c) может иметь силу только если  $c$  является accepted exhaustion certificate. TIMEOUT(t) сообщает о границе ресурса или времени. FAILURE(e) сообщает о failure-path. Ни один из двух последних outputs не подтверждает завершённый отрицательный поиск.

Протокол вправе объявить bridge:

if no response within 30 seconds, record operational status OFFLINE

Тогда по timeout выводится operational code OFFLINE. Это convention-input протокола. Из него не следует физическое отсутствие устройства, его уничтожение или невозможность позднего ответа.

Аналогично 404, zero, false, пустой array и negative sensor reading получают смысл только внутри exact protocol. Пустой array может быть успешным результатом total query, результатом фильтра, redaction, failure fallback или placeholder. Сами две скобки не выбирают interpretation.

## 10. Ракета: максимальная техническая уступка

Допускается полный application package:

- rocket-reference независимо связан с физическим объектом;
- formal calculation принят exact checker;
- calculation-to-design binding введён;
- controller действительно использовал вычисленные параметры;
- causal model принят;
- launch occurrence независимо установлен;
- telemetry complete относительно protocol;
- trajectory удовлетворяет заранее записанному criterion;
- ракета существует и летит.

Ни один пункт не отменяется. После максимальной уступки остаются разные вопросы:

$$q_{flight} \neq q_{goal} \neq q_{endorse} \neq q_{justify}.$$

Если calculation-record не содержит goal-code, следует только:

$$q_{goal} \notin \text{Contains}(\text{calculation-record}).$$

Не следует:

$$q_{no\_goal}.$$

Цель могла быть введена в mission brief, намерении участника, policy, другом record или вообще не иметь accepted binding. CBE-4 не разрешает выбрать один вариант из молчания calculation-record.

Обратный переход также запрещён. Строка

```
goal: Mars
```

предъявляет goal-code в document contract. Она не предъявляет Марс, полёт, желание, ценность или endorsement. Даже independently established institutional intention устанавливает intention-status, а не обязательность цели.

## 11. Goal, endorsement и SENI-4 как application corollary

CBE-4 даёт первую границу:

$$q_{endorse} \notin Cl_R(S) \not\Rightarrow q_{reject} \in Cl_R(S).$$

Не одобрено двусмысленно. Оно может означать отсутствие endorsement-record, наличие protected rejection-code, незавершённое решение или stable absence в completion-class. Эти statuses должны быть разведены.

SENI-4 добавляет application corollary, но не заменяет CBE-4. Даже если весь purpose-blind technical trace одинаков и criterion выполнен, две completions могут различаться только endorsement-coordinate:

$$(d, ENDORSE), (d, REJECT).$$

Одинаковый trace не выбирает один status. Технический success-code не является normative bridge. Если правило

$$Meets_c(x) \rightarrow q_{endorse(c)}$$

добавлено к  $R$ , endorsement-code формально выводится. Тогда origin различия находится в rule/base, а не в полёте как таковом.

Связь двух результатов точна:

- CBE-4: отсутствие endorsement-code не является rejection-code;
- SENI-4: наличие success-trace не выбирает endorsement-code;
- positive control: exhaustive declared family может установить stable absence конкретного code только относительно этого family.

## 12. Justification и отсутствие возражения

Следующие переходы не даны:

objection не записан  $\not\Rightarrow$  objection отсутствует,

найденное objection противоречиво  $\not\Rightarrow$  thesis justified,

альтернатива не найдена  $\not\Rightarrow$  альтернативы невозможны.

Противоречие в одном objection может отклонить именно этот formal derivation внутри exact theory. Чтобы из этого получить thesis, требуется exhaustive disjunction или отдельное inference rule. Иначе остаются неперечисленные alternatives.

Отсутствие citation не доказывает отсутствие predecessor. Отсутствие recorded dependency не доказывает независимое происхождение. Отсутствие найденной ошибки не доказывает безошибочность вне finite audit family. При exact checklist допустим более узкий status:

no listed defect was found in the declared finite checklist

Его нельзя сокращать до дефектов нет, не перенося hidden completion claim.

Justification-code может иметь finite support tree: каждый rule-node перечисляет required bases из текущего  $U = S \cup E$  и дочерние claim-premises; terminal claim-node имеет пустые premises и непустой required-base set. Это положительный formal object. Bases остаются отдельным sort и не становятся claim-members дерева или closure. Само отсутствие attack-code такой support tree не создаёт.

### 13. Сто, двести и триста лет

Пусть  $R_t$  — record до момента  $t$ . Для каждого конечного горизонта возможно:

$$q \notin R_t.$$

Из этого не следует:

$$\forall t' > t: q \notin R_{t'}.$$

Следующее расширение может ввести  $q$ . Сто лет без goal-code являются более длинным local record, а не universal absence. Двести лет без endorsement-code не вводят rejection. Триста лет успешных полётов не вводят за чем.

Если объявлен exact finite horizon

$$H = \{0, 1, \dots, T\}$$

и protocol действительно исчерпывает все positions  $H$ , допустим вывод:

$q$  отсутствует во всех declared records horizon  $H$ .

Он не относится к  $T+1$ , другому protocol или внешнему миру. Если exact theory доказывает universal formula по собственному induction rule, получается theorem-code внутри theory. Формула для всех времён не превращается в завершённый физический архив всех времён.

Длительность не является bridge. Добавление purpose-blind строк сохраняет purpose-blindness. Если на 301-м году появляется protected goal, endorsement или justification record, это новое введение, а не извлечение из предыдущего молчания.

## 14. Максимальные возражения

### Возражение 1: «Классическая математика доказывает отрицания»

Полностью допускается. Accepted proof of  $\neg_{\theta}\varphi$  является положительным proof-object для отдельного claim-code. CBE-4 отделяет такой proof от отсутствия proof of  $\varphi$ .

### Возражение 2: «Proof by contradiction устанавливает отсутствие»

Он устанавливает conclusion-code по exact discharge rule. Contradiction должен быть выведен, а не отсутствовать. Переход к внешнему отсутствию требует binding theory и target; формальная корректность не предъявляет названное.

### Возражение 3: «База данных использует closed-world assumption»

Closed-world policy допустима как explicit input. Тогда not present  $\rightarrow$  negative status является protocol rule. Результат относится к snapshot, schema, access rights и completeness policy. Policy не была записана самим отсутствующим row.

### Возражение 4: «Отсутствие сигнала является сигналом»

Только при channel contract: ожидаемое окно, исправность sensor, синхронизация, detection probability и rule интерпретации. Тогда silence-pattern становится положительным measurement record. Без contract silence совместимо с отсутствием события, failure sensor, packet loss и задержкой.

### Возражение 5: «Отсутствие evidence является evidence отсутствия»

Вероятностное обновление допустимо при detection model, prior и likelihood. Получается conditional probability-code. Логическое entailment внешнего отсутствия не появляется автоматически. Model и prior являются inputs.

### Возражение 6: «Поиск перебрал всё»

Это положительный control. Если всё означает exact finite  $E$ , проверены все  $\text{Max}(E)$ , closure total и certificate принят, CBE-4 даёт stable absence относительно  $E$ . Если хотя бы одно условие не предъявлено, остаётся local NOT-FOUND.

### Возражение 7: «Пустое множество и ноль уже являются отсутствием»

Они являются mathematical objects/codes по объявленному theory contract. Cardinality zero относительно exact set не является отсутствием названного вне set. Запись 0 rockets in registry относится к registry до отдельного completeness bridge.

**Возражение 8: «Отрицательный result code достаточно понятен»**

Понятность чтения является действием learned contract. Machine enum может быть строгим и полезным. Его строгость не позволяет спутать NEGATIVE, UNKNOWN, ERROR, TIMEOUT и NOT-APPLICABLE.

**Возражение 9: «Если ракета летит, цель очевидна»**

Очевидность является reading/endorsement policy. Один полёт совместим с исследованием, демонстрацией, перевозкой, оружием, ошибкой, ритуалом или несколькими целями. Даже exact declared purpose не устанавливает, почему его следует одобрить.

**Возражение 10: «Достаточно долгое повторение превращает отсутствие в факт»**

Каждый конечный record допускает позднее extension, если completion-class не закрыт. Долгая серия может изменить probability-code внутри модели, но не отменяет CBE-4. Exact closed horizon даёт только horizon-qualified result.

**15. Self-application**

Файл обязан пройти собственную границу.

Строка CBE-4 не является теоремой до reading contract и accepted derivation. Строка доказательство не становится proof-object. Отсутствие ошибки в этом файле не доказывает отсутствие ошибки. Отсутствие внешней ссылки не доказывает отсутствие сходной работы. Отсутствие возражения в разделе 14 не доказывает, что исчерпаны все возражения.

Этот файл не содержит physical rocket. Даже если application независимо связан с реальной ракетой, текст не содержит её цель, ценность или оправдание как свойства линий. Он содержит goal-, value- и justification-codes.

Если executable audit проверит конечный список требований и вернёт pass, корректный вывод ограничен этим списком, exact version и проверенными bytes. pass не является безусловной истиной и не гарантирует отсутствие дефекта вне audit family.

Если CBE-4 устанавливает

$$StableAbsent_E(q)$$

для некоторого  $q$ , результат остаётся relative to  $N, B, S, A, E, R$ . Он не вводит  $q^-$ , пока это не делает отдельный rule. Если позже family расширено, прежний verdict не распространяется автоматически на новое family.

Наконец, CBE-4 не отвечает, зачем принимать CBE-4. Endorsement исследования не следует из его formal success. Иначе application нарушило бы SENI-4 и собственное различие goal/endorsement.

## 16. Итоговый разрез

Допустимые положительные результаты:

- formal negation с accepted proof-object;
- proof by contradiction внутри exact theory;
- complement относительно exact carrier;
- NOT-FOUND с проверяемым exhaustion certificate;
- stable absence относительно конечного declared completion-class;
- minimal counterextensions при открытой возможности;
- условный operational negative status по explicit protocol.

Запрещённые подмены:

- blank  $\Rightarrow$  named absence;
- no proof found  $\Rightarrow$  proof of negation;
- timeout  $\Rightarrow$  false;
- failure  $\Rightarrow$  not found;
- missing  $q^+$   $\Rightarrow$  present  $q^-$ ;
- contradiction-code  $\Rightarrow$  внешняя пустота;
- complement без universe;
- no goal-code  $\Rightarrow$  no goal;
- no endorsement-code  $\Rightarrow$  rejection;
- technical success  $\Rightarrow$  justification;
- 100/200/300 лет records  $\Rightarrow$  вечное отсутствие или зачем.

Финальный разрез application:

*Локальное непоявление не является отрицанием.*

*Стабильное отсутствие требует объявленного completion-class.*

*Даже оно не создаёт названное отсутствие вне контракта.*

**Короткий нож Phase 4:** формула может вывести отрицание; пустое место ничего не выводит.

## 05. ВРАЖДЕБНАЯ ПРОВЕРКА И САМОПРИМЕНЕНИЕ PHASE 4

### ОТСУТСТВИЕ ЗАПИСИ НЕ ЯВЛЯЕТСЯ ЗАПИСЬЮ ОТСУТСТВИЯ

#### 0. Назначение и нулевая уступка словам

Этот файл пытается разрушить CBE-4, MCW-4 и BND-4 на их точной области. Строки отсутствуют, полно, максимально, закрыто, отрицательно, невозможно не принимаются за установленные отношения.

До назначения ролей предъявлены четыре пары:

$$K_4 = \{(0,0), (0,1), (1,0), (1,1)\}.$$

Первая координата равна нулю в двух парах, а вторая в них различается. Поэтому  $p=0$  не определяет  $n=1$ . Только отдельное ограничение, например

$$K_{\oplus} = \{(0,1), (1,0)\},$$

делает  $n=1-p$ . Поздние роли positive-code-present и negative-code-present не меняют этот объект.

Phase 4 допускает настоящий отрицательный результат: если  $q$  не входит ни в одну least closure точного объявленного family, stable absence внутри этого family установлено. Оно не становится отдельным negative-atom, физическим отсутствием или внешней полнотой.

#### 1. Exact object и admission gates

Пусть  $N$  — конечный sort protected claim-codes, а  $B$  — отдельный конечный sort protected base-ids;  $S \subseteq B$  — current protected bases;  $A = B \setminus S$  — все non-current candidate protected bases. Поэтому

$$B = S \cup A, S \cap A = \emptyset.$$

Пусть

$$\emptyset \neq E \subseteq P(A), \emptyset \in E,$$

— точное конечное family, ordered by inclusion. Admissible далее означает только membership в  $E$ .

Пусть  $R$  — конечное множество positive nonnull hyperrules. Каждый  $r \in R$  имеет точный тип

$$Prem(r) \subseteq N, ReqBase(r) \subseteq B, Head(r) \in N,$$

и удовлетворяет

$$|Prem(r)| + |ReqBase(r)| > 0.$$

Для  $U \subseteq B$  и  $X \subseteq N$ :

$$T_U(X) = X \cup \{ \text{Head}(r) : r \in R, \text{ReqBase}(r) \subseteq U, \text{Prem}(r) \subseteq X \},$$

$$Cl_R(U) = \mu T_U \text{ с итерацией от } X_0 = \emptyset, W_q(E) : \Leftrightarrow q \in Cl_R(S \cup E),$$

где  $q \in N$  зафиксирован заранее. Поэтому  $Cl_R(U) \subseteq N$ : base-ids из  $U$  проверяют required-base conditions, но не объединяются с closure и не становятся claim-members. Base-only introduction имеет  $\text{Prem}(r) = \emptyset$  и непустой  $\text{ReqBase}(r)$ . Также:

$$\text{Max}(E) = \{ M \in E : \nexists E \in E (M \subsetneq E) \},$$

$$CE_q = \{ E \in E : W_q(E) \}, \text{MinCE}_q = \text{Min}_{\subseteq} (CE_q).$$

Обязательные admission gates:

1. **FINITE-TYPED.** Все множества конечны; claims типизированы в  $N$ , bases — в отдельном  $B$ ;  $S, A \subseteq B$ ; premises и heads принадлежат  $N$ , required bases —  $B$ , а  $q \in N$ .
2. **EXACT-BASE-PARTITION.**  $A = B \setminus S$ , поэтому  $B = S \cup A$  и  $S \cap A = \emptyset$ ; каждый известный base-id имеет ровно одну current/candidate роль, один id не маскируется в обеих.
3. **CURRENT-MEMBER.**  $E \neq \emptyset$  и  $\emptyset \in E$ .
4. **EXACT-FAMILY.** Каждый  $E \subseteq A$ ; family канонизировано, дедуплицировано и заморожено.
5. **POSITIVE-NONNULL.** Для каждого rule  $|\text{Prem}| + |\text{ReqBase}| > 0$ ; нет deletion, default, defeat или absence-test. Пустые premises допустимы только при непустом required-base set.
6. **LEAST-CLOSURE.** Closure пересчитана от bottom  $X_0 = \emptyset$  при параметре bases  $U = S \cup E$ , а не получена объединением с  $U$ , взята из payload или заменена nonleast fixed point.
7. **FROZEN-BOUNDARY.**  $N, B, S, A, E, R, q$  зафиксированы до просмотра verdict.
8. **RECOMPUTED-ORDER.** Inclusion, maxima, CE и MinCE вычислены, а не приняты из claimed fields.
9. **VERDICT-EXHAUSTION.** Для stable-absence verdict проверены все exact maxima либо независимо доказан cofinal test-set. Timeout не проходит gate.
10. **MINCE-EXHAUSTION.** Для exact  $CE_q$ , полного antichain  $\text{MinCE}_q$  и MCW-cover пересчитан каждый member полного declared family; один cofinal frontier этого не устанавливает.
11. **PROTECTED-FIELDS.** Payload не создаёт base, rule, family member, target, polarity или completeness-policy.
12. **FAMILY-RELATIVE.** Итог относится только к exact tuple  $(N, B, S, A, E, R, q)$ .
13. **SEPARATE-POLARITY.**  $q^+$  и  $q^-$  — разные ids; complement-rule не подразумевается.

Нарушение gate выводит объект из theorem-domain; оно не является контрпримером.

## 2. CBE-4, MCW-4 и BND-4

### 2.1. CBE-4

$$StableAbsent_E(q) : \Leftrightarrow \forall E \in E \neg W_q(E).$$

CBE-4 утверждает:

$$StableAbsent_E(q) \Leftrightarrow \forall M \in Max(E) \neg W_q(M) \Leftrightarrow CE_q = \emptyset \Leftrightarrow MinCE_q = \emptyset.$$

Причина maximal reduction — не слово completion, а МОНОТОННОСТЬ:

$$E \subseteq E' \Rightarrow Cl_R(S \cup E) \subseteq Cl_R(S \cup E') \Rightarrow W_q(E) \leq W_q(E').$$

В конечном poset каждый member содержится в maximal member. Поэтому enabling member переносит  $q$  на maximal superset. CE пусто ровно при universal absence; конечное непустое CE обязательно имеет minimal element.

### 2.2. MCW-4

$$\neg StableAbsent_E(q) \Leftrightarrow MinCE_q \neq \emptyset.$$

Для каждого enabling  $E$  существует

$$E_* \in MinCE_q, E_* \subseteq E,$$

а  $q$  имеет конечный derivation witness: каждый rule-node перечисляет required base-ids из  $S \cup E_*$  и дочерние witnesses для claim-premises; terminal claim-node имеет пустые premises и непустой required-base set. Base-id подтверждает применимость rule, но не является claim-member closure.

Монотонность даёт и обратное включение, поэтому

$$CE_q = \{ E \in E : \exists E_* \in MinCE_q (E_* \subseteq E) \}.$$

Если дополнительно выполнено local-not-found  $\neg W_q(\emptyset)$ , каждый  $E_*$  непуст. Без этой посылки  $\emptyset$  может быть minimal witness текущего состояния.

Minimal означает inclusion-minimal enabling member объявленного family. Без downward closure оно не означает, что каждый atom  $E_*$  используется tree.

### 2.3. BND-4

Если два instances имеют одинаковые  $N, B, S, A, R, q$ , но

$$StableAbsent_{E_0}(q) \neq StableAbsent_{E_1}(q),$$

family-coordinate различает outputs и входит в readset. Процедура, не читающая различие  $E_0 / E_1$ , не выдаёт оба family-relative результата.

$$E = \{ \emptyset \}$$

— declared closed-instance, но не свидетельство внешней полноты.

### 3. Missing maxima и cofinality

Пусть claim-sort и base-sort заданы отдельно:

$$N = \{ q \}, B = \{ a, b \}, S = \emptyset, A = \{ a, b \},$$

$$E = \{ \emptyset, \{ a \}, \{ b \} \}, R = \{ r_a \},$$

$$ReqBase(r_a) = \{ a \}, Prem(r_a) = \emptyset, Head(r_a) = q.$$

Точные maxima:  $\{ a \}$  и  $\{ b \}$ . Проверка только  $\{ b \}$  даёт ложный negative verdict, хотя  $\{ a \} \in CE_q$ . Это MISSING-MAXIMAL-MEMBER, а не CBE-falsifier.

В конечном family любой cofinal test-set содержит все maxima: maximal member не имеет большего declared member. Claimed list maxima должен быть пересчитан. Фраза достаточно большие cases проверены не является cofinality certificate.

Если finiteness снята, maxima могут отсутствовать. Universal по пустому *Max* тогда вакуузен. Это OUTSIDE-CBE-4-NONFINITE-OR-NONCOFINAL.

### 4. Немонотонный target, default и defeat

Maximal reduction ложна для произвольного  $Q$ . Пусть

$$E = \{ \emptyset, \{ a \}, \{ a, b \} \},$$

$$Q(\emptyset) = 0, Q(\{ a \}) = 1, Q(\{ a, b \}) = 0.$$

Единственный maximal member имеет  $Q = 0$ , но middle member имеет  $Q = 1$ . Этот  $Q$  не является  $W_q$  positive least-closure: добавление  $b$  отменяет output.

Та же атака поражает maximal-check при negation-as-failure, defeater, exception, priority override, retraction и temporal deletion. Такие semantics могут быть точными, но требуют полной проверки family либо отдельной theorem. Поле monotone:true без доказательства не помогает.

CWA также допустима как отдельная policy. На одном поoccurrence core OWA может вернуть UNDETERMINED, CWA — negative-role code. Outputs различены policy, а не молчанием. Если default-

system скомпилирован в новый monotone state-space с explicit policy/stage codes, CBE применяется уже к новому object.

## 5. Incomplete search, omitted extension и timeout

CBE квантифицирует по exact  $E$ , не по просмотренному prefix. Для отрицательного результата требуется полный search или sound cofinal reduction.

Если scanner предыдущего control обработал  $\emptyset$  и  $\{b\}$ , но пропустил  $\{a\}$ , он не вычислил CBE. Нужно различать:

1. Member записан в  $E$ , но implementation его пропустила — implementation failure.
2. Input содержит меньшее  $E'$  — theorem может быть верна для  $E'$ , но BND фиксирует новую boundary.

Generator является input и TCB. Нужны termination, canonical enumeration и равенство generated set заявленному family. Duplicate collision, silently skipped invalid member, stale cache и list/hash/count mismatch снимают admission.

Допустимые run-results различны:

```
COUNTEREXTENSION-FOUND
EXHAUSTED-NONE
INCOMPLETE-SEARCH
TIMEOUT
RESOURCE-ERROR
INVALID-MEMBER
```

Один валидный witness опровергает StableAbsent. Для EXHAUSTED-NONE закрывается весь declared quantifier. Timeout и ничего пока не найдено не равны  $CE_q = \emptyset$ .

## 6. Post hoc boundary

Пусть

$$N = \{q\}, B = \{a\}, S = \emptyset, A = \{a\}, R = \{r_a\},$$

$$ReqBase(r_a) = \{a\}, Prem(r_a) = \emptyset, Head(r_a) = q.$$

Для

$$E_0 = \{\emptyset\}$$

StableAbsent истинно; для

$$E_1 = \{\emptyset, \{a\}\}$$

оно ложно и  $MinCE_q = \{\{a\}\}$ . Core одинаков, различает family.

Удалить  $\{a\}$  после нахождения witness, сузить  $A$ , заменить  $q$ , rule-set или admission predicate — значит создать новый instance. Exact hashes  $N, B, S, A, E, R, q$  и canonicalization фиксируются до проверки.

## 7. Empty, malformed и inconsistent family

При  $E = \emptyset$  universals и sets CE/MinCE пусты вакуозно, но current instance отсутствует. Поэтому обязательны  $E \neq \emptyset$  и  $\emptyset \in E$ . Непустое family без  $\emptyset$  также не содержит current state  $S$ .

Inconsistent не является встроенным свойством payload. Возможны разные случаи:

- $E \not\subseteq A$ : schema-invalid;
- member нарушает заранее записанный admission predicate: он не входит в frozen family;
- payload содержит строку inconsistent: строка инертна.

Нельзя найти counterextension и затем удалить его новым фильтром. Если фильтр оставляет пустое family или удаляет  $\emptyset$ , gate не пройден.

Codes  $q^+$  и  $q^-$  в одном claim-closure не являются противоречием без exact exclusivity constraint. Family member  $E \subseteq A \subseteq B$  содержит только base-ids, а не эти claim-codes. Точный record-case — BOTH-CODES-PRESENT.

## 8. MinCE и support-basis нельзя склеивать

Пусть  $N = \{q\}$ ,  $B = \{a, b\}$ ,  $S = \emptyset$ ,  $A = \{a, b\}$  и

$$E = \{\emptyset, \{a, b\}\}, R = \{r_a\},$$

$$ReqBase(r_a) = \{a\}, Prem(r_a) = \emptyset, Head(r_a) = q.$$

$\{a, b\}$  — единственный MinCE, хотя  $b$  не используется. Member  $\{a\}$  отсутствует в family. Поэтому ДОПУСТИМО minimal enabling declared extension, НО НЕ каждый candidate необходим.

Для

$$E = P(\{a, b\}), R = \{r_a, r_b\},$$

где у  $r_a$  и  $r_b$  пустые  $Prem$ , heads равны  $q$ , а required-base sets равны соответственно  $\{a\}$  и  $\{b\}$ ,

получаем

$$MinCE_q = \{\{a\}, \{b\}\}.$$

Все несравнимые witnesses сохраняются. Derivation witness с rule-level required bases из  $S \cup E_*$  не утверждает, что названия этих bases истинны вне contract или что каждый element  $E_*$  использован.

Если  $W_q(\emptyset)$ , то  $\emptyset \in MinCE_q$ . Реализация не должна требовать непустой delta.

## 9. Negative-code conflation

Для разных ids  $q^+, q^-$  допустимы четыре closure-cases: neither, positive only, negative only, both. CBE для  $q^+$  устанавливает отсутствие  $q^+$  во всех declared closures; оно не устанавливает присутствие  $q^-$ .

Чтобы перейти от nonoccurrence  $q^+$  к  $q^-$ , нужен точный добавленный объект:

- complement-rule;
- exact-one domain restriction;
- CWA policy;
- derivation с head  $q^-$ ;
- base-triggered introduction rule с head  $q^-$ .

Negative-code может быть выведен, но не является initial base: bases принадлежат  $B$ , а  $q^- \in N$ . Для появления  $q^-$  нужен rule с этим head и выполнимыми premises/required bases. Запрещено считать его уже данным из-за отсутствия  $q^+$ .

Checker verdict  $StableAbsent(q^+)$  также не является atom  $q^-$  в прежнем closure. Для embedding нужен новый rule и descendant computation.

## 10. Обязательные positive controls

### 10.1. Exhaustive negative control

Пусть

$$\begin{aligned} N &= \{c, q\}, B = \{a, b\}, S = \emptyset, A = \{a, b\}, \\ E &= P(\{a, b\}), R = \{r_c\}, \\ Prem(r_c) &= \{c\}, ReqBase(r_c) = \emptyset, Head(r_c) = q. \end{aligned}$$

Ни  $c$ , ни rule с head  $c$  не доступны. Поэтому на каждом declared extension  $q$  отсутствует:

$$StableAbsent_E(q), CE_q = \emptyset, MinCE_q = \emptyset.$$

Это действительный family-relative negative result без derivation  $q^-$ . Overclaim отрицательный вывод всегда требует отрицательной посылки ложен. Exclusion structure находится в exact family и rules.

## 10.2. Enabling controls

При  $N = \{q\}$ ,  $B = A = \{a, b\}$ ,  $S = \emptyset$  и base-only rule

$$ReqBase(r_a) = \{a\}, Prem(r_a) = \emptyset, Head(r_a) = q$$

family  $E = P(\{a, b\})$  даёт  $MinCE_q = \{\{a\}\}$ . Два rules с required-base sets  $\{a\}$  и  $\{b\}$  дают два witnesses; один rule с  $ReqBase = \{a, b\}$  и пустым  $Prem$  даёт  $\{a, b\}$ . Controls проверяют OR между rules и AND required bases внутри rule. Claims в  $Prem$  образуют отдельную AND-координату того же exact rule.

Система, которая принимает только negative fixture и отбрасывает валидное enabling extension, зашивает лозунг вместо CBE.

## 11. Mutable history и payload spoof

Порядок Phase 4 добавляет candidate bases при неизменных  $N, B, S, A, R, q$ . Удаление из  $S$ , replacement rule, изменение bytes под прежним id или backdating создают новый instance. Новый verdict не переписывает старый freeze.

Manifest, timestamp и signature являются records по собственному контракту; их имена не удостоверяют физическое время.

Payload strings

```
q is absent
negative:q
complete:true
all extensions checked
counterextension:none
closed world
```

не создают fields, не завершают search и не становятся certificate. Fake verdict и hidden rule внутри payload инертны: checker их хранит в record hash, но не исполняет и не обязан отвергать весь record. Exact duplicate id и alias current/candidate отвергаются; ids case-sensitive, поэтому различный регистр означает разные ids, а не неявную коллизию.

## 12. Rocket/export

Допускается полностью: ракета построена, расчёт использован, telemetry совпала с prediction, controller сработал, ракета летит и техническая цель достигнута.

Если telemetry-, event-, binding- или goal-claim-codes выводятся rules, они появляются в closure. Соответствующие base-ids могут входить в  $S$  или candidate  $E$  и лишь активировать такие rules; сами bases в closure не входят. Такой  $E$  может снять StableAbsent — это positive control.

Если enabling bases отсутствуют во всём declared family, CBE устанавливает только:

q-code absent from every declared least closure

Отсюда не следуют отсутствие ракеты, события, цели, смысла или любого будущего input. Обратно, успешный полёт не backdate-ит goal-claim в прежний  $Cl_R(S)$  и не помещает какой-либо base-id в прежний  $S$ . Family  $\{\emptyset\}$  говорит только, что в данном instance additions не рассматриваются.

### 13. Самоприменение

Этот файл — record. Его theorem-formulas используют supplied sets, rules, inclusion-order и least policy. Они не выводят собственную метатеорию из отсутствия альтернативного файла.

Family будущих редакций, содержащее только текущую версию, не доказывает невозможность критики. Поздняя редакция образует новый family-instance; BND локализует изменение.

Self-audit может проверить text, hashes, gates и fixtures. Он не удостоверяет внешнюю полноту  $E$ , payload или квантор заголовка. Phase 4 не утверждает невозможность отрицательных формальных теорем: § 10.1 принимает их внутри declared boundary.

### 14. Exact theorem falsifiers

Theorem-falsifier обязан сохранять все gates и использовать independently recomputed objects.

#### 14.1. CBE-4

CBE-4-FALSIFIER — exact finite  $(N, B, S, A, E, R, q)$  positive nonnull least-closure contract, где различаются truth-values хотя бы двух строк:

1.  $StableAbsent_E(q)$ ;
2. universal absence на  $Max(E)$ ;
3.  $CE_q = \emptyset$ ;
4.  $MinCE_q = \emptyset$ .

Nonmonotone target, infinite family, missing member или timeout не проходят gates.

#### 14.2. MCW-4

MCW-4-FALSIFIER возникает, если:

- StableAbsent ложно, но MinCE пусто;
- enabling  $E$  не содержит никакого  $E_* \in MinCE_q$ ;
- MinCE-member не является enabling;
- $q \in Cl_R(S \cup E_*)$ , но finite derivation witness с rule-level required\_base\_ids и premise-witnesses отсутствует при том же contract.

Неиспользуемый candidate внутри MinCE без меньшего family-member не является falsifier: MCW не заявляет leaf-essentiality.

### 14.3. BND-4

BND-4-FALSIFIER требует одинакового core  $N, B, S, A, R, q$ , двух families с разными exact StableAbsent-values и одной детерминированной процедуры, которая не читает family-различия, но выдаёт оба правильных результата на одинаковом core-input.

Разные outputs из-за clock, cache или hidden file — omitted-read implementation failure. Canonical-byte-identical families при одинаковом core не могут иметь разные mathematical values.

## 15. Implementation falsifiers

Реализация требует исправления, если она:

1. доверяет claimed closure, maxima, CE или MinCE;
2. пропускает maximal member или не доказывает cofinality;
3. использует maximal reduction для nonmonotone evaluator;
4. принимает default, defeat, deletion или negative premise как positive rule;
5. выдаёт timeout/partial scan за empty CE;
6. silently исключает declared member или включает незаписанный;
7. меняет boundary после verdict;
8. принимает пустое family, family без  $\emptyset$ ,  $E \notin A$  или  $S \cap A \neq \emptyset$ ;
9. склеивает  $q^+$  и  $q^-$  без rule;
10. читает negative/completeness из payload;
11. теряет один из нескольких MinCE;
12. называет каждый atom MinCE necessary leaf;
13. использует nonleast closure;
14. зависит от hidden order, network, clock, random seed или stale cache;
15. принимает mutable object под прежним freeze;
16. переносит family-relative status на referent, event, cause, purpose или мир.

Implementation defect не меняет theorem truth на admissible objects.

## 16. Claim registry и границы

| Claim                                                            | Gate                                  | Поражение                        |
|------------------------------------------------------------------|---------------------------------------|----------------------------------|
| CBE-4: четыре условия эквивалентны                               | finite family; monotone least closure | CBE-4-FALSIFIER                  |
| Maxima достаточно                                                | maxima exact и cofinal                | missing maximal / nonmonotone Q  |
| MCW-4: nonabsence iff MinCE nonempty                             | exact finite inclusion order          | MCW-4-FALSIFIER                  |
| Каждый enabling E содержит MinCE                                 | полное family                         | enabling E без contained witness |
| MinCE даёт finite derivation witness с required-base annotations | positive typed contract               | accepted q без такого witness    |
| MinCE — minimal declared extension                               | не essential-leaf claim               | § 8 control                      |
| BND-4: family различает status                                   | identical core                        | BND-4-FALSIFIER                  |
| $\{\emptyset\}$ externally complete                              | такого gate нет                       | REJECTED-COMPLETENESS-EXPORT     |
| StableAbsent(q+) создаёт q-                                      | complement-rule отсутствует           | NEGATIVE-CODE-CONFLATION         |
| Timeout означает absence                                         | exhaustion отсутствует                | TIMEOUT-AS-NEGATIVE              |
| Payload назначает status                                         | protected fields                      | PAYLOAD-BOUNDARY-SPOOF           |
| Rocket/world следует из family status                            | family-relative theorem               | REJECTED-SCOPE-EXPORT            |
| Документ самосертифицирован                                      | supplied metatheory                   | REJECTED-SELF-EXCEPTION          |

Phase 4 применена неверно, если current nonoccurrence перепутано со StableAbsent; StableAbsent — с  $q^-$ ; maximality — с внешней полнотой; mutation — с extension; либо formal negative theorem объявлена невозможной вопреки exhaustive control.

## 17. Финальная граница красной команды

После атак сохраняется точный результат:

*В finite exact family positive least closure монотонна. Поэтому q отсутствует во всех declared closures ровно тогда, когда оно отсутствует на всех exact maxima, CE пусто и MinCE пусто. Если stable absence ложно, существует minimal enabling declared extension и finite derivation witness. Family, меняющее verdict при одинаковом core, является различающим input.*

Одновременно:

- missing positive-code не является negative-code;

- unfinished search не является empty witness-set;
- maximal declared member не является внешней полнотой;
- CWA/default/defeat не являются молчаливой частью positive closure;
- negative-code может быть выведен по rules, но не возникает из nonoccurrence без rule/domain/policy;
- family-relative отрицательный результат не предъявляет событие или его отсутствие.

Короткая публичная строка:

*Не записано — не значит записано нет. Нет может быть выведено только в предъявленной границе; сама граница остаётся входом.*

Эта строка также остаётся carrier.