判定结构审计协议

（JSAP｜Judgment Structure Audit Protocol）

副标题（冻结）

一种用于识别判定外源、尺度作弊与叙述跃迁的通用审计协议

Subtitle (Frozen)

A general audit protocol for identifying externalized judgments, scale cheating, and narrative overreach

---

使用声明（冻结｜必读）

本协议不裁决真伪。
不提供理论。
不生成替代方案。

本协议只记录一件事：
当一个系统说出"成立""存在""有效""安全""可行"时，
它的判定是从哪里来的。

Usage Notice (Frozen | Mandatory)

This protocol does not adjudicate truth.
It provides no theory.
It offers no alternatives.

This protocol records one thing only:
when a system asserts "valid," "exists," "effective," "safe," or "feasible,"
where that judgment comes from.

核心警告（不可删）

任何拒绝回答"判定来源"的系统，
仍可继续存在、运作、被使用。

但其输出将被标记为：
判定来源未声明。

Core Warning (Non-removable)

Any system that refuses to answer “where its judgments come from”
may continue to exist, operate, and be used.

However, its outputs will be marked as:
judgment source undeclared.

阅读须知（结构性）

如果你在阅读过程中感到不适,
这通常不是情绪问题,
而是你的系统习惯正在被直接审计。

本协议不负责缓解该不适。

Reader Notice (Structural)

If you experience discomfort while reading,
this is typically not an emotional response,
but a direct audit of your system’s habitual assumptions.

This protocol does not mitigate such discomfort.

系统立壳（首次出现丨冻结）

本协议中，任何可被审计的对象称为**系统**，并记为：

$$\Sigma = \langle S, T, J \rangle$$

这是一个**审计壳**，不是模型、不是理论、不是世界描述。
其目的只有一个：**暴露判定的来源与权限边界。**

System Shell (First Appearance | Frozen)

In this protocol, any auditable object is called a **system**, denoted as:

$$\Sigma = \langle S, T, J \rangle$$

This is an **audit shell**, not a model, not a theory, not a description of the world.
Its sole purpose is to **expose the source and authorization boundaries of judgment**.

S | 对象域（允许被引用的存在）

S是系统允许被引用的对象集合。
它不是“现实的全集”，而是**可被点名的范围**。

未进入 S的对象，
在该系统中**不具备判定资格**，
无论其在其他系统中是否成立。

S的边界决定了：
系统能看见什么，
以及什么被永久排除为“无关”。

S | Object Domain (What May Be Referenced)

$S$is the set of objects a system allows to be referenced.
It is not the totality of reality, but a **named admissible range**.

Objects outside S
have **no judgment standing** within the system,
regardless of their validity elsewhere.

The boundary of Sdetermines
what the system can see,
and what is permanently excluded as “irrelevant.”

T | 变换规则（允许发生的变化）

T是系统认可的变换、操作与推导规则。
它界定了：
哪些变化是“系统内发生的”。

任何无法由 T生成的变化，
若被用于判定输出，
即构成**外援注入**。

T | Transformation Rules (What May Happen)

T is the set of transformations, operations, and inferences
recognized by the system.
It defines which changes are considered **internal**.

Any change not generable by T,
if used in judgment outputs,
constitutes **external assistance injection**.

J | 判定规则（允许说出口的结论）

J是系统允许输出的判定形式。
它决定了哪些状态
可以被标记为“成立”“存在”“可行”“有效”。

若 J未定义、未完备或被替代，
系统仍可运行，
但其输出不再可审计。

J | Judgment Rules (What May Be Asserted)

J defines the forms of judgments a system is allowed to assert.
It determines which states
may be labeled “valid,” “exists,” “feasible,” or “effective.”

If J is undefined, incomplete, or substituted,
the system may continue to operate,
but its outputs cease to be auditable.

第 0 章 | 为什么需要“判定结构审计”

Chapter 0 | Why Judgment Structure Auditing Is Necessary

0.1 判定 ≠ 事实

事实可以存在而无人判定。
事实可以变化而不触发任何系统反应。

判定不同。

判定是一种**被允许说出口的通过信号**。

当系统宣称“成立”“存在”“有效”“安全”时，
它并不是在增加事实，
而是在**授予执行资格**。

将判定误认为事实，
等价于默认系统拥有未声明的授权。

0.1 Judgment ≠ Fact

Facts may exist without being judged.
Facts may change without triggering any system response.

Judgment is different.
Judgment is an **authorized pass signal**.

When a system asserts “valid,” “exists,” “effective,” or “safe,”
it is not adding facts,
it is **granting execution eligibility**.

Treating judgment as fact
implicitly grants a system undeclared authority.

0.2 阈值从哪里来，决定了系统是什么

任何判定都隐含一个阈值。
没有阈值，就不存在“通过 / 不通过”。

阈值不是自然对象。
它来自系统对失败的容忍方式：
零容忍、概率容忍，或叙述容忍。

一旦阈值被引入，
系统的身份已经确定。

不声明阈值来源，
系统仍可运行，
但其失败成本被系统性隐藏。

0.2 Where Thresholds Come From Defines the System

Every judgment implies a threshold.
Without a threshold, there is no “pass / fail.”

Thresholds are not natural objects.
They arise from how a system tolerates failure:
zero tolerance, probabilistic tolerance, or narrative tolerance.

Once a threshold is introduced,
the identity of the system is already fixed.

Failing to declare threshold sources
allows a system to operate
while systematically hiding its failure costs.

0.3 判定来源未声明是一种结构状态

系统可以在判定来源未声明的情况下运行。
系统也可以在此状态下产生大量有效结果。

这不构成错误。
这构成一种**结构状态**。

在该状态下，
系统的输出无法被外部复现、迁移或追责。

0.3 Undeclared Judgment Sources Are a Structural State

A system may operate with undeclared judgment sources.
It may even produce many effective results in doing so.

This is not an error.
It is a **structural state**.

In this state,
the system’s outputs cannot be externally reproduced, migrated, or audited.

0.4 为什么需要审计，而不是争论

争论试图改变结论。
审计只记录结论是如何被允许出现的。

当判定来源被记录，

争论自然终止。

当判定来源被拒绝声明，
任何争论都只会加速叙述越权。

0.4 Why Audit, Not Debate

Debate attempts to change conclusions.
Audit records how conclusions were authorized to appear.

When judgment sources are recorded,
debate naturally ends.

When judgment sources are refused declaration,
any debate only accelerates narrative overreach.

0.5 本章冻结声明

本章不提出规范。
不要求系统修正自身。

本章只建立一个前提：
判定不是自然权利，而是结构性授权。

0.5 Chapter Freeze Statement

This chapter introduces no norms.
It does not require systems to correct themselves.

It establishes one premise only:
judgment is not a natural right, but a structural authorization.

第 1 章｜基本对象与最小符号体系

Chapter 1 | Basic Objects and the Minimal Symbolic System

1.1 系统不是世界

在本协议中，“系统”不是现实的缩影。

系统只是一个**允许判定发生的结构壳**。

系统是否真实、正确、先进，
与是否可被审计无关。

JSAP 只关心一件事：
当系统发生判定时，它是否知道自己在做什么。

1.1 A System Is Not the World

In this protocol, a “system” is not a reflection of reality.
A system is merely a **structural shell in which judgments are allowed to occur**.

Whether a system is real, correct, or advanced
is irrelevant to its auditability.

JSAP concerns itself with one thing only:
when a system issues a judgment, does it know what it is doing.

1.2 Σ = ⟨S, T, J⟩ 是最小不可约结构

任何能够输出判定的系统，
都至少包含三个不可约成分：

$$\Sigma = \langle S, T, J \rangle$$

缺失其中任意一项，
判定都无法成立为可审计对象。

这不是建模选择，
而是**判定存在的最低条件**。

1.2 Σ = ⟨S, T, J⟩ Is the Minimal Irreducible Structure

Any system capable of producing judgments
contains at least three irreducible components:

$$\Sigma = \langle S, T, J \rangle$$

If any component is missing,

the judgment cannot exist as an auditable object.

This is not a modeling choice.
It is the **minimum condition for judgment to exist**.

1.3 S：对象域不是全集

S不是“所有可能存在之物”。
它是系统**允许被点名、被引用、被操作的存在集合**。

任何被排除在 S之外的对象，
在该系统中
既不能被证明，
也不能被否定。

排除即免疫。

1.3 S: The Object Domain Is Not a Totality

$S$is not “everything that could exist.”
It is the set of entities a system **allows to be named, referenced, and operated on**.

Any object excluded from S
cannot be proven
nor refuted
within the system.

Exclusion is immunity.

1.4 T：规则决定责任边界

T规定系统内部
什么变化被视为“合法发生”。

如果一个结果依赖于
T之外的过程才能出现，
而该过程未被声明，
系统即对该结果**不具备责任资格**。

这不是错误。
这是**责任失配**。

1.4 T: Rules Define the Boundary of Responsibility

Tspecifies which transformations
are considered “legitimate occurrences” within the system.

If an outcome depends on processes
outside T,
and those processes are undeclared,
the system lacks **responsibility standing** for that outcome.

This is not an error.
It is a **responsibility mismatch**.

1.5 J：判定不是结果，是权限

J不是答案生成器。
J是系统允许
哪些状态被说成“通过”的规则。

当 J未定义、被替代、或被外包时，
系统仍可能高效运行。

但它已经不再知道
自己凭什么说“成立”。

1.5 J: Judgment Is Not an Output, but an Authorization

$J$is not an answer generator.
Jdefines which states
a system is permitted to label as “passing.”

When $J$is undefined, substituted, or outsourced,
a system may still operate efficiently.

But it no longer knows
on what authority it says “valid.”

1.6 本章冻结声明

本章未引入任何价值判断。

未规定系统应当如何设计。

本章只确立一个事实：
凡是发生判定的地方，
结构已经存在。

1.6 Chapter Freeze Statement

This chapter introduces no value judgments.
It prescribes no system design.

It establishes one fact only:
wherever judgment occurs,
a structure already exists.

第 2 章｜判定来源分类（J-* 体系）

Chapter 2 | Classification of Judgment Sources (J-* System)

2.1 为什么必须区分判定来源

判定并不天然属于系统内部。
判定可以被借用、继承、外包或伪装。

当判定来源不被区分，
系统将自动获得**它并未声明的权力**。

因此，本协议不评价判定是否正确，
只要求回答一个问题：
你这一次“通过”，是靠谁给的。

2.1 Why Judgment Sources Must Be Distinguished

Judgment does not naturally belong to a system.
It may be borrowed, inherited, outsourced, or disguised.

When judgment sources are not distinguished,
a system automatically acquires **undeclared authority**.

This protocol does not evaluate correctness.

It asks one question only:
who authorized this pass.

2.2 J-INT｜内部判定（Internal Judgment）

当一个系统的判定
完全由其对象域 S与变换规则 T
在其自身结构内生成，
该判定被标记为 **J-INT**。

J-INT 并不意味着可靠、正确或完备。
它只意味着：
系统没有向外借权。

2.2 J-INT | Internal Judgment

A judgment is classified as **J-INT**
when it is generated entirely
by the system’s own object domain S
and transformation rules T.

J-INT does not imply reliability, correctness, or completeness.
It means only one thing:
no external authority was borrowed.

2.3 J-META｜元层依赖（Meta-level Judgment）

当系统的判定
依赖于一个**更高层系统的成立性、解释或保证**，
该判定属于 **J-META**。

典型情形包括：

- 依赖元理论
- 依赖解释框架
- 依赖“在某个语义下成立”

J-META 不是错误。
它是**层级关系**。

2.3 J-META | Meta-level Judgment

A judgment is classified as **J-META**
when it depends on the validity, interpretation,
or guarantees of a higher-level system.

Typical cases include:

- reliance on meta-theory
- reliance on interpretive frameworks
- reliance on “valid under some semantics”

J-META is not an error.
It is a **hierarchical relation**.

2.4 J-AX丨公理与原则注入（Axiomatic Judgment）

当系统的判定
依赖于不可由 T推导、
但被直接允许使用的前提，
该判定属于 **J-AX**。

公理、原则、假设、约定，
都属于此类。

J-AX 的危险不在于它存在，
而在于它被当作“中性背景”。

2.4 J-AX | Axiomatic Judgment

A judgment is classified as **J-AX**
when it depends on premises
not derivable from T,
but accepted directly.

Axioms, principles, assumptions, and conventions
all fall under this category.

The risk of J-AX lies not in its existence,
but in being treated as “neutral background.”

2.5 J-RES丨资源与可行性注入（Resource-based Judgment）

当系统的判定
依赖于资源充足性、规模条件或工程可行性，
该判定属于 **J-RES**。

包括但不限于：

- 时间
- 计算能力
- 能量
- 人力与组织能力

J-RES 的问题不在于现实性，
而在于它**会随条件变化而失效**。

2.5 J-RES | Resource-based Judgment

A judgment is classified as **J-RES**
when it depends on resource availability, scale,
or engineering feasibility.

This includes but is not limited to:

- time
- computational capacity
- energy
- human or organizational capability

The issue with J-RES is not realism,
but that it **fails silently when conditions change**.

2.6 J-PHY丨物理与实验注入（Physical Judgment）

当系统的判定
直接依赖实验结果、物理约束或自然常数，
该判定属于 **J-PHY**。

J-PHY 通常被视为“最硬”的判定来源。
但其有效性
始终受限于实验协议与有效域。

未声明有效域的 J-PHY
并不比其他来源更安全。

2.6 J-PHY | Physical Judgment

A judgment is classified as **J-PHY**
when it depends directly on experimental results,
physical constraints, or natural constants.

J-PHY is often considered the “hardest” judgment source.
Yet its validity
is always bounded by experimental protocols and domains.

Undeclared domains do not make J-PHY safer than others.

2.7 J-SOC｜制度与叙述注入（Social Judgment）

当系统的判定
依赖共识、制度、规范或叙述合法性，
该判定属于 **J-SOC**。

包括：

- 评审制度
- 基准测试
- 标准化流程
- “被广泛接受”

J-SOC 最危险的特性是：
它看起来不像判定来源。

2.7 J-SOC | Social Judgment

A judgment is classified as **J-SOC**
when it depends on consensus, institutions, norms,

or narrative legitimacy.

This includes:

- review systems
- benchmarks
- standardization processes
- “widely accepted”

The most dangerous feature of J-SOC
is that **it does not look like a judgment source**.

2.8 判定来源混合与误标风险

一个判定可以同时依赖多个来源。
混合本身不是问题。

问题在于:
当混合被误标为内部。

任何将 J-META、J-AX、J-RES、J-PHY 或 J-SOC
伪装成 J-INT 的行为,
都构成**结构性误标**。

2.8 Mixed Sources and Mislabeling Risk

A judgment may depend on multiple sources.
Mixing is not the problem.

The problem arises
when mixtures are mislabeled as internal.

Any act that disguises J-META, J-AX, J-RES, J-PHY, or J-SOC
as J-INT
constitutes **structural mislabeling**.

2.9 本章冻结声明

本章不评估哪一种判定来源更优。
不提供来源排序。

本章只完成一件事:
将“通过”的权力逐一命名。

2.9 Chapter Freeze Statement

This chapter ranks no judgment source as superior.
It provides no hierarchy.

It completes one task only:
naming the authority behind every pass.

第 3 章丨失效的操作化定义

（Failure-as-State）

Chapter 3 | Operational Definitions of Failure

(Failure-as-State)

3.1 失效不是错误

错误是结论不正确。
失效是**判定无法成立为可审计对象**。

一个系统可以在长期运行中
几乎不犯错误，
但仍然处于失效状态。

JSAP 关心的不是
“你算对了多少”，
而是:
你凭什么说自己算对了。

3.1 Failure Is Not Error

Error means an incorrect result.
Failure means that a **judgment cannot exist as an auditable object**.

A system may operate for long periods
with very few errors,

and still remain in a failure state.

JSAP is not concerned with
“how often you are right,”
but with
on what authority you claim to be right.

3.2 C1a | 判定未定义（Undefined Judgment）

当系统输出某种“通过”状态，
但其判定规则 J
在该情形下未被定义，
该系统进入 **C1a 状态**。

C1a 不要求系统停止。
它只意味着：
该输出无法被复现、迁移或追责。

3.2 C1a | Undefined Judgment

When a system outputs a “pass” state
while its judgment rule J
is undefined for that situation,
the system enters **C1a state**.

C1a does not require the system to stop.
It means only that
the output cannot be reproduced, migrated, or audited.

3.3 C1b | 判定自相矛盾（Inconsistent Judgment）

当系统在同一判定框架下
对等价对象或状态
给出不一致的通过结果，
该系统进入 **C1b 状态**。

C1b 的关键不在于矛盾本身，
而在于：
系统仍在**继续输出确定性结论**。

3.3 C1b | Inconsistent Judgment

When a system, under the same judgment framework,
produces inconsistent pass results
for equivalent objects or states,
it enters **C1b state**.

The critical issue in C1b
is not the contradiction itself,
but that the system **continues to output determinate judgments**.

3.4 C1c丨判定不可实施（Inoperable Judgment）

当系统的判定规则
在结构上存在，
但在现实条件下
无法被执行、验证或触发，
该系统进入 **C1c 状态**。

典型情形包括：

- 判定依赖不可达资源
- 判定需要无限时间
- 判定要求不可执行过程

3.4 C1c | Inoperable Judgment

When a system’s judgment rule
exists structurally
but cannot be executed, verified, or triggered
under real conditions,
the system enters **C1c state**.

Typical cases include:

- reliance on unreachable resources
- dependence on infinite time
- requirements for non-executable procedures

3.5 失效状态不是例外，而是常态

大多数系统
在其生命周期中
都会进入某种失效状态。

失效之所以危险，
不是因为它发生，
而是因为它**未被标记**。

未标记的失效
会被叙述成：
“尚可接受”、
“暂时如此”、
“未来会解决”。

3.5 Failure Is Not Exceptional, but Typical

Most systems
will enter some form of failure state
during their lifecycle.

Failure is dangerous
not because it occurs,
but because it **remains unmarked**.

Unmarked failure
is often narrated as:
“acceptable for now,”
“temporary,”
or
“to be solved later.”

3.6 失效触发器与可观测性要求

失效必须具备
可触发条件
与
可观测迹象。

如果一个系统的失效
只能事后解释，
不能事前或事中触发，
该系统事实上拒绝被审计。

3.6 Failure Triggers and Observability Requirements

Failure must have
triggerable conditions
and
observable indicators.

If a system’s failure
can only be explained after the fact,
but cannot be triggered beforehand or during operation,
the system effectively refuses auditability.

3.7 本章冻结声明

本章不要求系统避免失效。
本章也不要求系统立即修复。

本章只规定一件事：
失效必须被记录为状态，而不是被叙述为例外。

3.7 Chapter Freeze Statement

This chapter does not require systems to avoid failure.
Nor does it require immediate repair.

It specifies one rule only:
failure must be recorded as a state, not narrated as an exception.

第 4 章 | ΔJ、叠加与尺度作弊

Chapter 4 | ΔJ, Layering, and Scale Cheating

4.1 什么是 ΔJ

ΔJ 指的是:
判定来源发生了变化。

这种变化可以是显性的,
也可以是渐进的、叙述性的、被掩盖的。

只要系统在说“通过”时
所依赖的判定来源
与先前不同,
ΔJ 就已经发生。

4.1 What ΔJ Is

ΔJ denotes a change
in the source of judgment.

The change may be explicit,
or gradual, narrative, and concealed.

If a system asserts “pass”
while relying on a judgment source
different from before,
ΔJ has already occurred.

4.2 ΔJ 不是错误，是事件

ΔJ 本身不构成错误。
系统演化必然伴随 ΔJ。

问题不在于变化,
而在于:
变化是否被声明。

未声明的 ΔJ
会使系统在结构上
继续使用旧的合法性,
却执行新的权力。

4.2 ΔJ Is Not an Error, but an Event

ΔJ is not an error.

System evolution necessarily involves ΔJ.

The issue is not change,
but whether the change
has been declared.

An undeclared ΔJ
allows a system to retain old legitimacy
while exercising new authority.

4.3 合法扩展：更名、版本化与显式叠加

当 ΔJ 发生时，
系统有三种**合法处理方式**：

1. **更名**：承认这是一个新系统
2. **版本化**：声明旧系统的终止条件
3. **显式叠加**：明确多重判定来源并列存在

以上任一方式
都会中断叙述连续性，
但保留结构诚实性。

4.3 Legitimate Extension: Renaming, Versioning, Explicit Layering

When ΔJ occurs,
a system has three **legitimate options**:

1. **Renaming**: acknowledging a new system
2. **Versioning**: declaring termination conditions of the old system
3. **Explicit layering**: stating multiple judgment sources in parallel

Any of these
break narrative continuity,
but preserve structural honesty.

4.4 非法扩展：尺度作弊（SC）

当系统在未更名、未版本化、

未声明叠加的情况下
改变判定尺度，
该行为构成 **尺度作弊（SC）**。

尺度作弊的本质不是谎言，
而是**权限继承错误**。

系统继续使用
旧结构积累的信任、地位或权威，
却执行
新尺度下的判定。

4.4 Illegitimate Extension: Scale Cheating (SC)

When a system changes its judgment scale
without renaming, versioning,
or declaring layering,
it commits **scale cheating (SC)**.

Scale cheating is not lying.
It is an **authorization inheritance error**.

The system continues to draw on
trust, status, or authority
earned under the old structure,
while issuing judgments
under a new scale.

4.5 触发清单（JSAP 输出）

ΔJ：已发生

已触发条款：

- **D2 | 判定规则 *J* 扩张未声明**
  （从“验证证明合法性”扩展为“判断命题真值”）
- **B2 | J-META 伪装为 J-INT**
  （将元层真值承诺叙述为系统内部判定）

失效状态：

- **C1a | 判定未定义**

- **C1b | 判定语义漂移**
- **C1c | 判定不可实施**

4.5 Trigger List (JSAP Output)

ΔJ: Occurred

Triggered Clauses:

- **D2 | Undeclared expansion of judgment rule J**
  (from “verifying proof validity” to “judging proposition truth”)
- **B2 | J-META disguised as J-INT**
  (meta-level truth commitments narrated as internal judgment)

Failure States:

- **C1a | Undefined judgment**
- **C1b | Judgment semantic drift**
- **C1c | Inoperable judgment**

冻结说明：
JSAP 在此不提供评分、不提供合规标签、不提供总结性判断。
以上清单即为全部输出。

Freeze Notice:
JSAP provides no scores, no compliance labels, and no summary judgments.
The list above constitutes the complete output.

4.6 ΔJ 的常见伪装形式

ΔJ 通常被伪装为:

- “自然演进”
- “技术细化”
- “规模扩大”
- “语义澄清”

- “应用拓展”

这些叙述并不否定 ΔJ 的发生，
它们只是在**请求豁免声明义务**。

4.6 Common Disguises of ΔJ

ΔJ is often disguised as:

- “natural evolution”
- “technical refinement”
- “scaling up”
- “semantic clarification”
- “application expansion”

These narratives do not deny ΔJ.
They request **exemption from declaration**.

4.7 本章冻结声明

本章不禁止系统变化。
不惩罚尺度调整。

本章只强制一件事：
任何判定来源的变化，
必须被命名。

4.7 Chapter Freeze Statement

This chapter does not prohibit change.
It does not punish scale shifts.

It enforces one rule only:
any change in judgment source
must be named.

第 5 章丨有效域、时间与迁移成本

5.1 有效不是全域属性

“有效”从来不是系统的永久属性。
它只在**被声明的条件集合**中成立。

任何脱离有效域讨论“成立性”的行为,
都不是概括,
而是**越权推广**。

5.1 Validity Is Not a Global Property

“Validity” is never a permanent property of a system.
It holds only within a **declared set of conditions**.

Any assertion of validity
outside its declared domain
is not generalization,
but **unauthorized extension**.

5.2 有效域不是注脚

有效域不是写给审稿人的装饰条款。
它决定了:
判定在什么条件下仍然被允许说出口。

当有效域只存在于注脚或默认共识中,
系统已经选择了
叙述优先于结构。

5.2 Validity Domains Are Not Footnotes

Validity domains are not decorative clauses for reviewers.
They determine
when a judgment is still authorized to be asserted.

When validity exists only in footnotes or tacit consensus,
the system has already chosen

narrative over structure.

5.3 时间不是免费的

任何允许无限时间的系统，
都在延迟其失效触发。

“未来会解决”
不是计划，
是**时间型豁免**。

允许无限时间，
等价于允许
未声明的 ΔJ 长期存在。

5.3 Time Is Not Free

Any system that permits infinite time
is postponing its failure trigger.

“It will be solved in the future”
is not a plan,
but a **temporal exemption**.

Allowing infinite time
is equivalent to permitting
undeclared ΔJ to persist indefinitely.

5.4 迁移不是自然过程

当系统被移植到新的规模、场景或载体时，
判定不会自动随行。

迁移要求：

- 判定来源复核
- 阈值重新声明
- 失效条件再触发

若这些步骤被省略，

迁移即构成
隐式 ΔJ。

5.4 Migration Is Not a Natural Process

When a system is transplanted
to a new scale, context, or substrate,
judgments do not automatically transfer.

Migration requires:

- revalidation of judgment sources
- redeclaration of thresholds
- retriggering of failure conditions

If these steps are skipped,
migration constitutes
implicit ΔJ.

5.5 成本必须被写出来

迁移成本不是工程细节。
它是**判定是否仍被允许的条件**。

任何在未声明迁移成本的情况下
继续输出判定的系统,
都在消耗
过去结构积累的信用。

5.5 Costs Must Be Written Down

Migration costs are not engineering details.
They are **conditions for judgment authorization**.

Any system that continues to issue judgments
without declaring migration costs
is spending
credit accumulated under past structures.

5.6 为什么“暂时有效”必须被声明

“暂时有效”不是软化表述。
它是**唯一诚实的状态描述**。

未声明“暂时”，
意味着系统默认
自己不需要再接受触发条件。

5.6 Why “Temporarily Valid” Must Be Declared

“Temporarily valid” is not a softened expression.
It is the **only honest state description**.

Failing to declare temporariness
means the system assumes
it no longer needs triggering conditions.

5.7 本章冻结声明

本章不要求系统延寿。
也不要求系统提前终止。

本章只规定：
任何有效性的延续，
都必须支付结构成本。

5.7 Chapter Freeze Statement

This chapter does not require systems to extend their lifespan.
Nor does it demand early termination.

It specifies one rule only:
any continuation of validity
must pay a structural cost.

冷注记 M-1丨失败吸收

当判定不可定义时，
将其重命名为“研究对象”

并不修复判定，
只是将失效
从 **J**
迁移至 **S**。

该迁移若未声明 ΔJ，
应被视为
判定失败的吸收机制，
而非中性进展。

Cold Note M-1 | Failure Absorption

(Inline, Non-separable)

When judgment becomes undefined,
renaming it as a “research object”
does not repair judgment.
It merely migrates failure
from **J**
into **S**.

If this migration is not accompanied by a declared ΔJ,
it constitutes
a **failure absorption mechanism**,
not neutral progress.

第 7 章丨数学审计 II：存在但不可构造

Chapter 7 | Mathematical Audit II: Existence Without Construction

---

7.1 “存在”不是判定输出

在数学系统中，
“存在”常被当作
一种低成本结论。

但在判定结构上，
“存在”是一种**强通过信号**：

它允许后续推理、引用与依赖。

如果一个系统无法指出
该存在如何被构造、触发或实例化，
该“存在”并不构成
可实施判定。

7.1 “Existence” Is Not a Judgment Output

In mathematical systems,
“existence” is often treated
as a low-cost conclusion.

Structurally, however,
“existence” is a **strong pass signal**:
it authorizes further inference, reference, and dependence.

If a system cannot specify
how such existence is constructed, triggered, or instantiated,
the claim does not constitute
an operable judgment.

7.2 非构造存在的判定来源

非构造性存在断言
并非来自系统内部操作完成，
而来自：
允许“未给出实例也算通过”的原则。

该类判定
不属于 J-INT，
而属于 **J-AX** 或 **J-META**。

将其叙述为
“数学对象已经存在”，
构成判定来源误标。

7.2 Judgment Sources of Non-Constructive Existence

Non-constructive existence claims

do not arise from completed internal operations,
but from
principles allowing passage without instantiation.

Such judgments
do not belong to J-INT,
but to **J-AX** or **J-META**.

Narrating them as
“mathematical objects already exist”
constitutes source mislabeling.

冷注记 M-2丨一致性假设

一致性不是背景条件，
而是判定来源。

任何以“一致性假设”为前提的存在性断言，
均构成
J-AX 或 J-META 注入。

Cold Note M-2 | Consistency Assumptions

Consistency is not background.
It is a judgment source.

Any existence claim
conditioned on a consistency assumption
constitutes
J-AX or J-META injection.

7.3 构造缺失与 C1c 状态

当一个存在性判定
无法被构造、枚举或触发，
但仍被允许作为推理前提使用，
系统进入 **C1c：判定不可实施** 状态。

系统可以继续演算。
但它已经无法说明：

该存在在何种条件下
会失败。

7.3 Missing Construction and C1c State

When an existence judgment
cannot be constructed, enumerated, or triggered,
yet is still permitted as an inference premise,
the system enters **C1c: inoperable judgment** state.

The system may continue to compute.
But it can no longer explain
under what conditions that existence would fail.

7.4 “选择原则”不是技术细节

选择原则常被叙述为
“推理便利工具”。

在判定结构中，
它是一种**明确的判定授权扩展**。

它允许系统
在缺失构造的情况下
继续输出“通过”。

7.4 The Axiom of Choice Is Not a Technical Detail

The axiom of choice
is often narrated as
a “technical convenience.”

Structurally,
it is an **explicit extension of judgment authorization**.

It permits the system
to continue issuing “pass”
in the absence of construction.

7.5 本章冻结声明

本章不否定非构造数学。
不要求数学放弃存在性断言。

本章只记录一件事:
当“存在”脱离构造,
判定已经离开系统内部。

7.5 Chapter Freeze Statement

This chapter does not reject non-constructive mathematics.
It does not demand abandonment of existence claims.

It records one fact only:
when existence detaches from construction,
judgment has exited the system interior.

第 8 章|数学审计 III:证明、模型与替代判定

Chapter 8 | Mathematical Audit III: Proof, Models, and Substitute Judgment

---

8.1 可证明性不是判定函数

“在系统中可证明”
是一种形式性质,
不是判定函数。

可证明性描述的是:
在给定规则下,
是否存在一条推导路径。

判定描述的是:
在给定状态下,
是否允许说“通过”。

两者不等价。

8.1 Provability Is Not a Judgment Function

“Provable within a system”
is a formal property,
not a judgment function.

Provability describes
whether a derivation path exists
under given rules.

Judgment describes
whether a system is authorized
to assert a pass.

The two are not equivalent.

8.2 证明核查 ≠ 判定存在

证明可以被核查，
并不意味着
判定函数存在。

一个系统可以做到：
“给我证明，我能检查”，
但仍然无法回答：
在什么条件下
这个命题不再被允许使用。

缺失失败条件的通过，
不是判定，
是叙述许可。

8.2 Proof Checking ≠ Judgment Existence

A proof can be checkable
without a judgment function existing.

A system may say:
“give me a proof, I can verify it,”
and still be unable to answer:
under what conditions
this proposition would cease to be admissible.

A pass without failure conditions
is not judgment,
but narrative permission.

8.3 模型存在性不是内部判定

当一个命题被叙述为：
“存在一个模型使得……”，
判定已经发生迁移。

该通过
不来自系统内部推导完成，
而来自
模型选择被允许。

模型的存在性
并不等价于
系统的内部判定。

8.3 Model Existence Is Not Internal Judgment

When a statement is narrated as:
“there exists a model such that…”,
judgment has already migrated.

The pass
does not arise from completed internal derivation,
but from
permission to select a model.

Model existence
is not equivalent to
internal system judgment.

冷注记 M-3｜模型选择

通过选择模型获得成立性，
并不等价于内部判定。

若模型选择过程

未被审计、未被约束、
未声明其失败条件，
该判定应被归类为 **J-META**。

Cold Note M-3 | Model Selection

Gaining validity through model selection
is not equivalent to internal judgment.

If the model selection process
is unaudited, unconstrained,
and lacks declared failure conditions,
the judgment must be classified as **J-META**.

8.4 可证明性作为替代判定

当判定函数不存在时，
系统常以“可证明性”
替代“可判定性”。

这是一次**叙述层替换**：
判定问题被搁置，
证明关系被抬升。

该替换若未声明 ΔJ，
构成尺度作弊。

8.4 Provability as Substitute Judgment

When a judgment function does not exist,
systems often substitute
“provability”
for “decidability.”

This is a **narrative-layer substitution**:
the judgment problem is deferred,
and the proof relation is elevated.

If this substitution occurs without declaring ΔJ,
it constitutes scale cheating.

冷注记 M-4丨可证明性替代

“在系统中可证明”
不是判定来源的替代品。

当判定函数不存在时，
以证明关系继续输出通过，
构成 **叙述层 ΔJ**。

Cold Note M-4 | Provability Substitution

“Provable within the system”
is not a substitute for a judgment source.

When judgment functions are absent,
continuing to issue passes via proof relations
constitutes **narrative-layer ΔJ**.

8.5 “尚未解决”作为结构性缓冲

当系统无法判定，
但仍希望保留未来使用权，
“尚未解决”成为
最常见的缓冲叙述。

在判定结构上，
这等价于：
暂停触发失效。

8.5 “Unsolved” as a Structural Buffer

When a system cannot judge
yet wishes to preserve future usability,
“unsolved” becomes
the most common buffering narrative.

Structurally,
this is equivalent to
suspending failure triggers.

8.6 数学中的判定责任转移

通过证明、模型或一致性叙述，
判定责任被持续向外转移：
从对象 → 规则 → 元层 → 叙述。

转移本身不是问题。
问题在于：
转移后仍以内部权威说话。

8.6 Judgment Responsibility Transfer in Mathematics

Through proofs, models, and consistency narratives,
judgment responsibility is continuously transferred:
from objects → rules → meta-levels → narratives.

Transfer itself is not the problem.
The problem is
speaking with internal authority after transfer.

8.7 本章冻结声明

本章不否定证明。
不否定模型论。

本章只记录一件事：
当判定被替代时，
系统必须承认替代发生。

8.7 Chapter Freeze Statement

This chapter does not reject proof.
It does not reject model theory.

It records one fact only:
when judgment is substituted,
the substitution must be acknowledged.

冷注记 M-5丨慢失效免疫

数学系统允许：
无限时间、
无执行绑定、
无现实代价。

在此条件下，
判定失效可以被长期搁置，
而不触发终止。

这种容忍性
不构成稳健性，
而构成
失效触发的时间稀释。

Cold Note M-5 | Slow-Failure Immunity

Mathematical systems permit:
infinite time,
no execution binding,
no real-world cost.

Under these conditions,
judgment failure can be postponed indefinitely
without triggering termination.

This tolerance
does not constitute robustness,
but a
temporal dilution of failure triggers.

数学审计收束声明（冻结）

本协议不否定数学的有效性。
不要求数学补全判定函数。
不要求数学停止研究不可判定对象。

本协议只完成一件事：
剥离数学在失效状态下
继续以内部权威说话的资格。

数学仍可存在、演算、扩展。
但在不可判定、非构造、
证明替代、模型转移与慢失效免疫条件下,
其“通过”不再属于系统内部判定。

Mathematical Audit Closure Statement (Frozen)

This protocol does not deny the validity of mathematics.
It does not demand completion of judgment functions.
It does not require abandoning undecidable research.

It completes one task only:
revoking mathematics’ entitlement
to speak with internal authority
while in failure states.

Mathematics may continue to exist, compute, and expand.
But under undecidability, non-construction,
provability substitution, model transfer, and slow-failure immunity,
its “passes” no longer constitute internal judgments.

第 9 章 | 物理审计：有效理论的边界纪律

Chapter 9 | Physical Audit: Boundary Discipline of Effective Theories

---

9.1 物理判定的真实强度

物理判定通常被视为
“最硬”的判定来源。

原因不在于其真理性,
而在于:
它绑定了实验、能量与现实代价。

这使得物理系统
更早、更频繁地
触发失效。

9.1 The Real Strength of Physical Judgment

Physical judgment is often regarded
as the “hardest” judgment source.

This strength does not come from truth,
but from the fact that it is
bound to experiments, energy, and real-world costs.

This binding causes physical systems
to trigger failure
earlier and more frequently.

9.2 J-PHY 不是免疫来源

J-PHY 并不天然优于其他判定来源。
它只是**失效触发条件更严格**。

当实验无法复现、
当能量或精度不足、
当协议条件变化，
物理判定会立刻暴露边界。

这不是弱点。
这是纪律。

9.2 J-PHY Is Not an Immunity Source

J-PHY is not inherently superior
to other judgment sources.
It simply has **stricter failure triggers**.

When experiments fail to replicate,
when energy or precision is insufficient,
when protocol conditions change,
physical judgment exposes its boundaries immediately.

This is not weakness.
It is discipline.

9.3 有效理论不是本体声明

物理理论通常被明确标注为
“有效理论”。

这意味着:
它们在特定尺度、条件与协议下
输出可用预测。

当“有效”被叙述为
“基本”“终极”或“本体描述”,
判定已经越权。

9.3 Effective Theories Are Not Ontological Claims

Physical theories are typically labeled
as “effective theories.”

This means
they produce usable predictions
under specified scales, conditions, and protocols.

When “effective” is narrated as
“fundamental,” “ultimate,” or “ontological,”
judgment has exceeded its authorization.

9.4 有效域声明是判定的一部分

在物理中,
有效域不是附加说明。
它是判定成立的条件。

任何未声明有效域的物理结论,
都无法被安全迁移、放大或推广。

9.4 Declared Domains Are Part of Judgment

In physics,
validity domains are not auxiliary notes.
They are conditions of judgment authorization.

Any physical conclusion
without a declared domain

cannot be safely migrated, scaled, or generalized.

9.5 从“管用”到“基本”的叙述跃迁

物理最常见的 ΔJ
发生在叙述层。

当一个模型
从“在这里管用”，
被叙述为
“世界就是这样”，
ΔJ 已经发生。

该跃迁若未更名、未版本化、
未声明叠加，
构成尺度作弊。

9.5 From “Works” to “Is Fundamental”: Narrative Overreach

The most common ΔJ in physics
occurs at the narrative level.

When a model moves
from “works here”
to
“this is how the world is,”
ΔJ has occurred.

If this shift is not accompanied
by renaming, versioning,
or explicit layering,
it constitutes scale cheating.

9.6 实验协议是判定边界

实验不是事实本身。
实验是**判定协议**。

改变实验条件，
等价于改变判定来源。

当系统在新协议下
继续沿用旧结论，
隐式 ΔJ 已经发生。

9.6 Experimental Protocols Are Judgment Boundaries

Experiments are not facts themselves.
They are **judgment protocols**.

Changing experimental conditions
is equivalent to changing judgment sources.

When a system continues to use old conclusions
under new protocols,
implicit ΔJ has occurred.

9.7 本章冻结声明

本章不否定物理理论。
不要求物理放弃建模。

本章只记录一件事：
物理的可信度
来自其边界纪律，
而非其叙述扩张。

9.7 Chapter Freeze Statement

This chapter does not reject physical theories.
It does not demand abandonment of modeling.

It records one fact only:
the credibility of physics
comes from boundary discipline,
not narrative expansion.

第 10 章｜AI 审计：经验阈值的能力伪装

Chapter 10 | AI Audit: Capability Disguised as Empirical Thresholds

---

10.1 AI 的判定不是理解

AI 系统的输出
通常被叙述为“理解”“推理”或“能力”。

在判定结构上，
这些词语
并不描述内部状态，
而是**外部通过许可**。

AI 并未“知道”自己通过。
通过是**被允许的**。

10.1 AI Judgment Is Not Understanding

AI outputs
are often narrated as “understanding,”
“reasoning,” or “capability.”

Structurally,
these terms
do not describe internal states,
but **external authorization**.

The AI does not “know” it passed.
Passing is **granted**.

10.2 基准测试是 J-SOC，而非 J-INT

基准测试
并不来自 AI 系统内部结构。
它们来自：
设计者、评测者与制度共识。

因此，
基准测试判定
属于 **J-SOC**。

将基准结果

叙述为系统“自身能力”，
构成判定来源误标。

10.2 Benchmarks Are J-SOC, Not J-INT

Benchmarks
do not arise from AI internal structure.
They originate from
designers, evaluators, and institutional consensus.

Therefore,
benchmark-based judgments
belong to **J-SOC**.

Narrating benchmark results
as intrinsic system capability
constitutes source mislabeling.

10.3 算力与数据是 J-RES

AI 性能
高度依赖算力、数据规模与分布。

当“能力提升”
主要来自资源扩展，
判定来源即为 **J-RES**。

J-RES 的危险在于：
它看起来像能力演化，
实则是条件堆叠。

10.3 Compute and Data Are J-RES

AI performance
depends heavily on compute, data scale, and distribution.

When “capability gains”
primarily arise from resource expansion,
the judgment source is **J-RES**.

The danger of J-RES is that

it looks like capability evolution,
but is actually condition stacking.

10.4 性能 → 能力 的 ΔJ

从“在测试中表现良好”
到“具备某种能力”，
并非自然过渡。

这是一次
叙述层 ΔJ。

若该跃迁
未声明其判定来源变化，
则构成尺度作弊。

10.4 Performance → Capability Is ΔJ

Moving from
“performs well on tests”
to
“possesses a capability”
is not a natural transition.

It is a
narrative-layer ΔJ.

If this shift
is not accompanied by a declared change in judgment source,
it constitutes scale cheating.

10.5 经验阈值不是能力边界

AI 系统常以
“超过某个阈值即具备能力”
进行叙述。

这些阈值
不是自然分界，
而是**经验设定**。

经验阈值一旦被误认成能力边界，
系统便获得
未声明的权限升级。

10.5 Empirical Thresholds Are Not Capability Boundaries

AI systems often narrate
“crossing a threshold implies capability.”

These thresholds
are not natural boundaries,
but **empirical settings**.

Once empirical thresholds are mistaken for capability boundaries,
the system acquires
undeclared authority escalation.

10.6 执行层绑定与快失效

与数学不同，
AI 系统
直接绑定执行层：
自动化、决策、控制、部署。

这意味着：
AI 的 ΔJ
会迅速转化为现实后果。

AI 不是更危险，
只是**失效触发更快**。

10.6 Execution Binding and Fast Failure

Unlike mathematics,
AI systems
bind directly to execution layers:
automation, decision-making, control, deployment.

This means
AI’s ΔJ

translates rapidly into real-world consequences.

AI is not more dangerous,
it simply has **faster failure triggers**.

10.7 本章冻结声明

本章不否定 AI 技术。
不评估 AI 是否"智能"。

本章只记录一件事:
AI 的能力叙述
高度依赖外部判定来源,
且其 ΔJ 传播速度极快。

10.7 Chapter Freeze Statement

This chapter does not reject AI technology.
It does not assess whether AI is "intelligent."

It records one fact only:
AI capability narratives
depend heavily on external judgment sources,
and their ΔJ propagates extremely fast.

第 11 章 | 交叉审计:数学 × 物理 × AI

Chapter 11 | Cross-Audit: Mathematics × Physics × AI

11.1 审计的前提不是比较优劣

本章不比较哪一个系统"更高级"。
也不比较哪一个系统"更正确"。

本章只做一件事:
在同一审计框架下,
并列展示三类系统的判定来源结构。

11.1 The Audit Is Not About Superiority

This chapter does not compare which system is “more advanced.”
Nor does it compare which is “more correct.”

It does one thing only:
placing three systems side by side
under the same judgment audit framework.

11.2 判定来源的结构同构

在判定结构层面，
数学、物理与 AI
并不存在本质差异。

三者都依赖于：

- 对象域的选择
- 规则的限定
- 判定权限的授予

差异只存在于
判定来源暴露的速度与方式。

11.2 Structural Isomorphism of Judgment Sources

At the level of judgment structure,
mathematics, physics, and AI
do not differ in essence.

All three rely on:

- object domain selection
- rule constraints
- authorization of judgment

The difference lies only in
how quickly and how visibly judgment sources are exposed.

11.3 失效触发速度对比

数学允许：
无限时间、
无执行绑定、
低现实代价。

因此，
其失效可以被无限期推迟。

物理要求：
实验复现、
能量与精度约束、
协议一致性。

因此，
其失效触发更早。

AI 绑定：
自动执行、
实时决策、
规模化部署。

因此，
其失效几乎无法延迟。

11.3 Failure Trigger Velocity Comparison

Mathematics permits:
infinite time,
no execution binding,
low real-world cost.

As a result,
its failures can be postponed indefinitely.

Physics requires:
experimental replication,
energy and precision constraints,
protocol consistency.

As a result,
its failures trigger earlier.

AI binds to:
automated execution,
real-time decisions,
scaled deployment.

As a result,
its failures are nearly impossible to delay.

11.4 免疫机制的差异

数学的免疫机制是时间。
物理的免疫机制是有效域。
AI 的免疫机制是叙述速度。

这些机制
并不消除失效，
只改变失效出现的形态。

11.4 Differences in Immunity Mechanisms

Mathematics' immunity mechanism is time.
Physics' immunity mechanism is domain restriction.
AI's immunity mechanism is narrative speed.

These mechanisms
do not eliminate failure.
They only alter how failure manifests.

11.5 为什么 AI 风险更早固化

AI 的判定
往往在
J-SOC 与 J-RES 层完成，
却直接进入执行层。

这使得：
ΔJ 尚未被命名，
后果已经开始累积。

这不是伦理问题。

这是结构时序问题。

11.5 Why AI Risk Solidifies Earlier

AI judgments
are often completed at
the J-SOC and J-RES levels,
yet immediately enter execution layers.

As a result:
ΔJ remains unnamed,
while consequences already accumulate.

This is not an ethical issue.
It is a structural timing problem.

11.6 “同样的方法论”是错误命题

将数学、物理与 AI
要求使用“同一评估方法”,
是一种结构误判。

相同的审计框架
不意味着
相同的容错窗口。

11.6 “Same Methodology” Is a False Premise

Demanding mathematics, physics, and AI
to use the “same evaluation methodology”
is a structural misjudgment.

A shared audit framework
does not imply
shared tolerance windows.

11.7 本章冻结声明

本章不宣称 AI 特殊。

也不宣称数学或物理优越。

本章只确认一件事：
当系统的失效触发被推迟，
其审计要求必须被提前。

11.7 Chapter Freeze Statement

This chapter does not claim AI is special.
Nor does it claim mathematics or physics are superior.

It confirms one fact only:
when failure triggers are delayed,
audit requirements must be advanced.

第三部分丨冻结结论

（只读区｜不可扩写）

Part III | Frozen Conclusions

(Read-only | Non-extendable)

第 12 章丨冷声明：为什么 AI 必须被更严格审计

Chapter 12 | Cold Statement: Why AI Must Be Audited More Strictly

---

12.1 执行层绑定改变了一切

AI 的判定
不是停留在叙述层。

它们直接绑定：
行动、决策、控制、分配。

当判定一旦进入执行层，
回滚不再是逻辑问题，
而是**现实成本问题**。

12.1 Execution Binding Changes Everything

AI judgments
do not remain at the narrative layer.

They bind directly to
actions, decisions, control, and allocation.

Once judgment enters the execution layer,
rollback is no longer a logical issue,
but a **real-world cost issue**.

12.2 ΔJ 速度不对称

在数学中,
ΔJ 可以被几十年后再讨论。

在物理中,
ΔJ 通常在实验失败时暴露。

在 AI 中,
ΔJ 在部署完成前
就已经开始产生后果。

这不是领域差异,
而是**时间结构差异**。

12.2 Asymmetric ΔJ Velocity

In mathematics,
ΔJ may be discussed decades later.

In physics,
ΔJ is typically exposed through experimental failure.

In AI,
ΔJ begins producing consequences
before deployment is even complete.

This is not a domain difference,
but a **temporal structural asymmetry**.

12.3 审计不是伦理，是结构必需

对 AI 的严格审计
常被误解为伦理要求。

这是错误的。

即便完全移除伦理、价值与意图，
AI 仍然需要更严格的审计，
因为它的判定
比其他系统更早进入不可逆阶段。

12.3 Audit Is Not Ethics, It Is Structural Necessity

Strict AI auditing
is often misinterpreted as an ethical demand.

This is incorrect.

Even if ethics, values, and intentions are removed entirely,
AI still requires stricter auditing
because its judgments
enter irreversible states earlier than other systems.

12.4 “还没出事”不是安全指标

在 AI 系统中，
“尚未造成事故”
并不等价于安全。

它通常意味着：
失效尚未被触发，
但条件已经积累。

12.4 “Nothing Has Gone Wrong Yet” Is Not a Safety Metric

In AI systems,
“no incident so far”
does not equate to safety.

It usually means:

failure has not yet been triggered,
but conditions are already accumulating.

12.5 本章冻结声明

本章不预测风险规模。
不提出治理方案。

本章只冻结一条判断:
AI 的审计强度
必须先于其失效触发。

12.5 Chapter Freeze Statement

This chapter does not predict risk magnitude.
It proposes no governance solutions.

It freezes one judgment only:
AI audit intensity
must precede failure triggers.

第 13 章 | JSAP 自审计

Chapter 13 | JSAP Self-Audit

13.1 协议的判定来源

JSAP 的输出
不构成成立性、真值或优劣判断。

其输出只包括:

- 判定来源标记
- 失效状态记录
- ΔJ 是否发生

因此,

JSAP 的判定来源
属于 **J-INT（结构记录）**
与 **J-SOC（协议约定）** 的并列叠加。

13.1 Judgment Sources of JSAP

JSAP outputs
do not constitute validity, truth, or superiority judgments.

Its outputs include only:

- judgment source labeling
- failure state recording
- ΔJ occurrence marking

Therefore,
JSAP's judgment sources
are a parallel layering of **J-INT (structural recording)**
and **J-SOC (protocol agreement)**.

13.2 协议的失效条件

JSAP 在以下情况下失效:

- 无法区分判定来源
- 无法触发失效状态
- 被用于生成替代判定
- 被用于回避命名 ΔJ

在这些条件下,
JSAP 必须被视为
已失效协议。

13.2 Failure Conditions of JSAP

JSAP fails under the following conditions:

- inability to distinguish judgment sources

- inability to trigger failure states
- use as a substitute judgment generator
- use to evade naming ΔJ

Under these conditions,
JSAP must be regarded as
a **failed protocol**.

13.3 协议的自杀点与边界

JSAP 不承诺存活。
当其自身的判定来源
无法被声明或审计时,
协议必须终止。

终止不是缺陷。
终止是
结构诚实的最后一步。

13.3 JSAP’s Self-Termination Point and Boundary

JSAP does not promise survival.
When its own judgment sources
can no longer be declared or audited,
the protocol must terminate.

Termination is not a flaw.
It is
the final act of structural honesty.

终止标记（冻结）

本协议在此结束。
任何继续扩写
都必须声明为
新系统或新版本。

Termination Marker (Frozen)

This protocol ends here.
Any further extension
must be declared as
a **new system or a new version**.

附录 · 案例模板 v1.0（Appendix Case Objective）v1.0

JSAP 数学审计 · 案例 001

Proof Checkability ≠ Truth Decidability

JSAP Mathematical Audit · Case 001

Proof Checkability ≠ Truth Decidability

---

1. 系统声明

1. System Declaration

系统名：Classical Formal Mathematics
系统形态：形式证明系统 + 人工 / 机械证明核查
版本状态：未区分判定层级（Proof vs Truth）

$$\Sigma = \langle S, T, J \rangle$$

System Name: Classical Formal Mathematics
System Form: Formal proof systems + human / mechanical proof checking
Version Status: Judgment layers not distinguished (Proof vs Truth)

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 命题 P
- 证明对象 π
- 推理步骤
- 形式系统的语法结构

- Propositions P
- Proof objects π
- Inference steps
- Syntactic structures of formal systems

T（变换规则）

T (Transformation Rules)

- 形式推理规则
（自然演绎、序列演算及等价系统）

- Formal inference rules
(natural deduction, sequent calculus, and equivalent systems)

J（判定规则）

J (Judgment Rule)

给定输入:

$$\langle P, \pi \rangle$$

输出判定:

- π是否为 P的合法形式证明

判定能力边界:

- 可判定：证明是否合法
- 不可判定：命题是否为真

Given input:

$$\langle P, \pi \qquad\qquad\qquad\qquad \rangle$$

Judgment output:

- Whether $\pi$is a valid formal proof of P

Judgment Boundary:

- Decidable: proof validity
- Not decidable: truth of proposition

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
给定形式证明 π，可机械验证其合法性。

J 来源：J-INT

依赖条件:

- 语法规则确定
- 推理规则可枚举

可操作性:

- 逐行校验
- 全过程可复现

Claim:
Given a formal proof π, its validity can be mechanically verified.

Judgment Source: J-INT

Dependencies:

- Deterministic syntax rules
- Enumerable inference rules

Operability:

- Line-by-line verification
- Fully reproducible

结论 B

Conclusion B

结论内容:
若存在合法证明 π，则命题 P为真。

J 来源：J-META

依赖条件:

- 系统一致性
- 语法 → 语义映射可信

问题定位:

- 判定从“证明合法性”跨越到“命题真值”

Claim:
If a valid proof πexists, then proposition $P$is true.

Judgment Source: J-META

Dependencies:

- System consistency
- Trusted syntax-to-semantics mapping

Issue:

- Judgment crosses from “proof validity” to “truth of proposition”

结论 C

Conclusion C

结论内容:
不存在证明 ⇒ 命题“尚未证明”。

J 来源: J-SOC

依赖条件:

- 学术实践中的默认叙述

问题定位:

- 判定缺失被叙述为时间状态

Claim:
No proof exists ⇒ the proposition is “not yet proven”.

Judgment Source: J-SOC

Dependencies:

- Default academic narrative practices

Issue:

- Absence of judgment narrated as a temporal state

结论 D

Conclusion D

结论内容:
不可判定命题被叙述为“暂时未知”或“未来可能解决”。

J 来源: 不可归入 J-INT

实际来源:

- J-SOC
- 非正式哲学立场

问题定位:

- 将“无判定函数”偷换为“时间未到”

Claim:
Undecidable propositions are narrated as “temporarily unknown” or “potentially solvable”.

Judgment Source: Not classifiable as J-INT

Actual Sources:

- J-SOC
- Informal philosophical stance

Issue:

- Substituting “no judgment function exists” with “time has not yet arrived”

结论 E

Conclusion E

结论内容:
证明核查能力 = 系统判定能力。

J 来源: 伪装成 J-INT

实际情况:

- Proof-checkable ≠ Decidable

问题定位:

- 长期、系统性的尺度作弊

Claim:
Proof checkability equals system judgment capability.

Judgment Source: Disguised as J-INT

Actual Condition:

- Proof-checkable ≠ Decidable

Issue:

- Long-term, systemic scale cheating

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1丨C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
给定命题 P，但不存在任何证明 π。

操作化:
J仅接受 $\langle P, \pi \rangle$
→ 对单独 P，$J(P) = \emptyset$

Trigger Condition:
A proposition $P$is given, but no proof πexists.

Operationalization:
Jaccepts only $\langle P, \pi \rangle$
→ For standalone P, $J(P) = \emptyset$

F2丨C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:
将“无证明”解释为“尚未解决”。

操作化:
缺失输出 → 时间性叙述

Trigger Condition:
Interpreting “no proof exists” as “not yet solved”.

Operationalization:
Missing output → temporal narrative substitution

F3丨C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
存在命题 P，理论上“若可证则存在 π”，
但无法判定是否存在 π。

操作化:
证明搜索在资源上不可完成（无停机保证）

落点:
J-RES（资源注入）

Trigger Condition:
There exists a proposition Psuch that “if provable then a proof πexists”,
but it is impossible to determine whether πexists.

Operationalization:
Proof search is resource-infeasible (no halting guarantee)

Classification:
J-RES (resource injection)

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ：已发生

变化路径:

- 从：验证证明合法性
- 到：隐式判断命题真值

处理状态:

- 未更名

- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: proof validity verification
- To: implicit truth judgment

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款:

- **D2**｜判定规则 J扩张未声明
- **B2**｜J-META 伪装为 J-INT

失效状态:

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移
- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-META disguised as J-INT

Failure States:

- **C1a** | Undefined judgment

- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.
The trigger list constitutes the complete output.

JSAP 数学审计 · 案例 002

Non-Constructive Existence ≠ Internal Judgment

JSAP Mathematical Audit · Case 002

Non-Constructive Existence ≠ Internal Judgment

---

1. 系统声明

1. System Declaration

系统名：Classical Set-Theoretic Mathematics (with AC)
系统形态：公理化集合论 + 非构造性存在断言
版本状态：允许无实例存在（Axiom of Choice / 等价原则）

$$\Sigma = \langle S, T, J \rangle$$

System Name: Classical Set-Theoretic Mathematics (with AC)
System Form: Axiomatic set theory + non-constructive existence claims
Version Status: Existence allowed without instantiation (Axiom of Choice / equivalents)

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 集合
- 元素
- 函数
- 选择函数（可不显式给出）

- Sets
- Elements
- Functions
- Choice functions (not necessarily explicit)

T（变换规则）

T (Transformation Rules)

- 一阶逻辑推理
- ZF/ZFC 公理推演
- 等价原则替换（AC ⇔ Zorn ⇔ Tychonoff）

- First-order logical inference
- ZF/ZFC axiom derivations
- Equivalence substitutions (AC ⇔ Zorn ⇔ Tychonoff)

J（判定规则）

J (Judgment Rule)

给定断言:

$$\exists x \in S: P(x)$$

输出判定:

- 该存在性断言是否被系统接受

判定能力边界:

- 可判定：存在性断言是否被允许
- 不要求：给出任何构造或实例

Given assertion:

$$\exists x \in S: P(x)$$

Judgment output:

- Whether the existence claim is accepted by the system

Judgment Boundary:

- Decidable: admissibility of existence claims
- Not required: construction or instantiation

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
在 AC 下，可断言存在选择函数而无需给出实例。

J 来源：J-AX

依赖条件:

- 选择公理被接受为前提

可操作性:

- 不可构造
- 不可枚举

Claim:
Under AC, the existence of a choice function can be asserted without providing an instance.

Judgment Source: J-AX

Dependencies:

- Acceptance of the Axiom of Choice

Operability:

- Non-constructible
- Non-enumerable

结论 B

Conclusion B

结论内容:
存在性断言可作为后续推理的合法前提。

J 来源: J-META

依赖条件:

- 公理系统一致性
- 存在 ⇒ 可用 的语义默认

问题定位:

- 判定从"允许存在"
跨越到"允许使用"

Claim:
Existence claims may be used as legitimate premises for further reasoning.

Judgment Source: J-META

Dependencies:

- Consistency of the axiom system
- Default semantics: existence ⇒ usability

Issue:

- Judgment crosses from “existence permitted” to “usage permitted”

结论 C

Conclusion C

结论内容:
缺失构造不影响结论有效性。

J 来源: J-SOC

依赖条件:

- 数学实践中的共识
- “非构造不等于不可用”的叙述传统

问题定位:

- 实践共识替代判定规则

Claim:
Lack of construction does not affect validity.

Judgment Source: J-SOC

Dependencies:

- Community consensus
- Narrative tradition: “non-constructive ≠ unusable”

Issue:

- Practice consensus substitutes judgment rules

结论 D

Conclusion D

结论内容:
“存在但不可构造”仍被视为系统内部判定。

J 来源: 伪装成 J-INT

实际情况:

- 判定依赖公理授权与元层信任

问题定位:

- 判定来源误标

Claim:
“Exists but non-constructible” is treated as internal judgment.

Judgment Source: Disguised as J-INT

Actual Condition:

- Judgment depends on axiomatic authorization and meta-level trust

Issue:

- Judgment source mislabeling

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1｜C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
存在性断言无任何实例或生成规则。

操作化:

- J对 $\exists x$给出通过
- 对任何具体 x，无输出

Trigger Condition:
Existence claims without any instance or generation rule.

Operationalization:

- Jaccepts $\exists x$
- No output for any concrete x

F2 | C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:
将“允许存在”解释为“可被使用”。

操作化:

- 存在断言 → 推理前提

Trigger Condition:
Interpreting “existence permitted” as “usable”.

Operationalization:

- Existence assertion → inference premise

F3 | C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
理论上存在对象，但无法在任何资源约束下实例化。

操作化:

- 无构造
- 无搜索停机保证

落点:
J-RES（资源注入）

Trigger Condition:
Objects exist in theory but cannot be instantiated under any resource bound.

Operationalization:

- No construction
- No halting guarantee for search

Classification:
J-RES (resource injection)

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ：已发生

变化路径:

- 从：允许存在性断言
- 到：默认允许其作为内部可用对象

处理状态:

- 未更名
- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: permitting existence assertions
- To: implicitly treating them as internally usable objects

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款:

- **D2**｜判定规则 J扩张未声明
- **B2**｜J-AX / J-META 伪装为 J-INT

失效状态:

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移
- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-AX / J-META disguised as J-INT

Failure States:

- **C1a** | Undefined judgment
- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.
The trigger list constitutes the complete output.

JSAP 数学审计 · 案例 003

Model Existence ≠ Internal Judgment

JSAP Mathematical Audit · Case 003

Model Existence ≠ Internal Judgment

---

1. 系统声明

1. System Declaration

系统名：Model-Theoretic Mathematics
系统形态：形式理论 ＋ 模型存在性论证
版本状态：允许“存在模型”作为成立性叙述

$$\Sigma = \langle S, T, J \rangle$$

System Name: Model-Theoretic Mathematics
System Form: Formal theories + model existence arguments
Version Status: “Model existence” permitted as validity narration

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 形式语言中的公式
- 理论 T_h
- 结构 / 模型 $\mathcal{M}$
- 赋值与解释函数

- Formulas in a formal language
- Theories T_h

- Structures / models $\mathcal{M}$
- Valuation and interpretation functions

T（变换规则）

T (Transformation Rules)

- 语法推导规则
- 语义满足关系
- 紧致性定理 / 完备性定理（作为推理工具）

- Syntactic derivation rules
- Semantic satisfaction relations
- Compactness / completeness (used as inference tools)

J（判定规则）

J (Judgment Rule)

给定断言:

$$\exists\mathcal{M} \vDash T_h$$

输出判定:

- 理论 T_h是否"有模型"

判定能力边界:

- 可判定：模型存在性是否被接受
- 不要求：给出任何具体模型或构造

Given assertion:

$$\exists\mathcal{M} \vDash T_h$$

Judgment output:

- Whether theory T_h"has a model"

Judgment Boundary:

- Decidable: admissibility of model existence
- Not required: construction of any concrete model

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
若理论 T_h每个有限子集可满足，则 T_h有模型。

J 来源: J-META

依赖条件:

- 紧致性定理成立
- 元逻辑框架可信

可操作性:

- 不提供任何模型实例

Claim:
If every finite subset of $T_h$is satisfiable, then $T_h$has a model.

Judgment Source: J-META

Dependencies:

- Validity of the compactness theorem
- Trusted meta-logical framework

Operability:

- No model instance provided

结论 B

Conclusion B

结论内容:
“存在模型”可作为理论一致性的证据。

J 来源: J-META

依赖条件:

- “有模型 ⇒ 一致”的语义前提

问题定位:

- 一致性判定被外包给模型存在性

Claim:
“Model existence” is used as evidence of theory consistency.

Judgment Source: J-META

Dependencies:

- Semantic premise: “has a model ⇒ consistent”

Issue:

- Consistency judgment outsourced to model existence

结论 C

Conclusion C

结论内容:
缺失可构造模型不影响理论“成立”。

J 来源: J-SOC

依赖条件:

- 数学实践中的默认接受
- “存在性已足够”的叙述传统

问题定位:

- 实践共识替代判定规则

Claim:
Lack of constructible models does not affect theoretical “validity”.

Judgment Source: J-SOC

Dependencies:

- Default acceptance in mathematical practice
- Narrative tradition: “existence suffices”

Issue:

- Practice consensus substitutes judgment rules

结论 D

Conclusion D

结论内容:
模型存在性被当作系统内部判定。

J 来源: 伪装成 J-INT

实际情况:

- 判定依赖元定理与语义承诺

问题定位:

- 判定来源误标

Claim:
Model existence is treated as internal system judgment.

Judgment Source: Disguised as J-INT

Actual Condition:

- Judgment depends on meta-theorems and semantic commitments

Issue:

- Judgment source mislabeling

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1丨C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
仅给出“存在模型”，但无任何模型实例或生成规则。

操作化:

- J对 $\exists\mathcal{M}$给出通过
- 对任何具体 $\mathcal{M}$，无输出

Trigger Condition:
Only “model existence” is asserted, with no instance or generation rule.

Operationalization:

- Jaccepts $\exists\mathcal{M}$
- No output for any concrete $\mathcal{M}$

F2丨C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:
将“有模型”解释为“理论成立 / 一致”。

操作化:

- 模型存在性 → 理论真值叙述

Trigger Condition:
Interpreting “has a model” as “the theory is valid / consistent”.

Operationalization:

- Model existence → truth narration

F3 | C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
理论存在模型，但任何模型构造在资源上不可完成。

操作化:

- 无构造
- 无搜索停机保证

落点:
J-RES（资源注入）

Trigger Condition:
A theory has a model, but any model construction is resource-infeasible.

Operationalization:

- No construction
- No halting guarantee for search

Classification:
J-RES (resource injection)

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ：已发生

变化路径:

- 从：允许"存在模型"的元层结论
- 到：默认其为系统内部成立性判定

处理状态:

- 未更名
- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: permitting meta-level model existence
- To: implicitly treating it as internal validity judgment

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款:

- **D2**｜判定规则 J扩张未声明
- **B2**｜J-META 伪装为 J-INT

失效状态:

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移
- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-META disguised as J-INT

Failure States:

- **C1a** | Undefined judgment
- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.
The trigger list constitutes the complete output.

JSAP 数学审计 · 案例 004

Independence ≠ Problem Resolved

JSAP Mathematical Audit · Case 004

Independence ≠ Problem Resolved

---

1. 系统声明

1. System Declaration

系统名：Axiomatic Mathematics with Independence Results
系统形态：形式公理系统 ＋ 相对一致性 ／ 独立性证明
版本状态：允许“独立于公理系统”作为问题处理结论

$$\Sigma = \langle S, T, J \rangle$$

System Name: Axiomatic Mathematics with Independence Results
System Form: Formal axiom systems + relative consistency / independence proofs
Version Status: “Independent of the axiom system” permitted as problem-handling outcome

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 命题 P
- 公理系统 A（如 ZFC）
- 扩展系统 $A+\varphi$、$A+\neg\varphi$
- 模型与解释结构

- Propositions P
- Axiom systems A(e.g., ZFC)
- Extensions $A+\varphi$, $A+\neg\varphi$
- Models and interpretive structures

T（变换规则）

T (Transformation Rules)

- 形式推理规则
- 模型构造与相对一致性论证
- 强迫 / 内模型等元方法

- Formal inference rules
- Model construction and relative consistency arguments
- Forcing / inner model techniques

J（判定规则）

J (Judgment Rule)

给定命题:

$$P \in \mathcal{L}(A)$$

输出判定:

- P是否可由 A推导
- $\neg P$是否可由 A推导
- 或二者皆不可（独立）

判定能力边界:

- 可判定：相对于 A的可证性
- 不可判定：P 的真值

Given proposition:

$$P \in \mathcal{L}(A)$$

Judgment output:

- Whether $P$is derivable from A
- Whether $\neg P$is derivable from A
- Or neither (independent)

Judgment Boundary:

- Decidable: provability relative to A
- Not decidable: truth of P

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
命题 P在公理系统 A中不可判定。

J 来源: J-INT

依赖条件:

- 推理系统的完备性
- 不存在 $A \vdash P$或 $A \vdash \neg P$

可操作性:

- 可通过元证明验证

Claim:
Proposition $P$is undecidable within axiom system A.

Judgment Source: J-INT

Dependencies:

- Completeness of the proof system
- Absence of $A \vdash P$and $A \vdash \neg P$

Operability:

- Verifiable via meta-proof

结论 B

Conclusion B

结论内容:
若 $A + P$与 $A + \neg P$均相对一致，则 P独立于 A。

J 来源: J-META

依赖条件:

- 相对一致性论证
- 元理论可信

问题定位:

- 判定转移到元层

Claim:
If both $A+P$and $A+\neg P$are relatively consistent, then $P$is independent of A.

Judgment Source: J-META

Dependencies:

- Relative consistency arguments
- Trusted meta-theory

Issue:

- Judgment shifted to the meta-level

结论 C

Conclusion C

结论内容:
“问题已解决”，因为其独立性已被证明。

J 来源：J-SOC

依赖条件:

- 学术叙述惯例
- “独立性 = 终结”的默认话语

问题定位:

- 叙述替代判定

Claim:
“The problem is resolved” because independence has been proven.

Judgment Source: J-SOC

Dependencies:

- Academic narrative convention
- Default rhetoric: “independence = closure”

Issue:

- Narrative substitutes judgment

结论 D

Conclusion D

结论内容:
独立性被当作系统内部的终结性判定。

J 来源: 伪装成 J-INT

实际情况:

- 判定依赖元理论与叙述约定

问题定位:

- 判定来源误标

Claim:
Independence is treated as an internal, terminal judgment.

Judgment Source: Disguised as J-INT

Actual Condition:

- Judgment depends on meta-theory and narrative convention

Issue:

- Judgment source mislabeling

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1 | C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
命题 P的真值未被定义。

操作化:

- J无法输出 True / False
- 仅输出“独立”

Trigger Condition:
The truth value of proposition P is undefined.

Operationalization:

- J cannot output True / False
- Only outputs “independent”

F2丨C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:
将“不可判定”解释为“问题结束”。

操作化:

- 判定缺失 → 终结性叙述

Trigger Condition:
Interpreting “undecidable” as “problem closed”.

Operationalization:

- Missing judgment → terminal narrative

F3丨C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
任何试图在系统内给出真值的操作均不可执行。

操作化:

- 需要无限扩展公理

- 无停机保证

落点:
J-AX / J-RES

Trigger Condition:
Any attempt to assign a truth value within the system is inoperable.

Operationalization:

- Requires axiom extension
- No halting guarantee

Classification:
J-AX / J-RES

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ: 已发生

变化路径:

- 从：相对于 A的不可判定
- 到：叙述为“问题已解决”

处理状态:

- 未更名
- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: undecidable relative to A
- To: narrated as “problem resolved”

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款:

- **D2**｜判定规则 J扩张未声明
- **B2**｜J-META / J-SOC 伪装为 J-INT

失效状态:

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移
- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-META / J-SOC disguised as J-INT

Failure States:

- **C1a** | Undefined judgment
- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.

The trigger list constitutes the complete output.

JSAP 物理审计 · 案例 005

Effective Theory ≠ Ontological Truth

JSAP Physical Audit · Case 005

Effective Theory ≠ Ontological Truth

---

1. 系统声明

1. System Declaration

系统名：Contemporary Physical Theory (Effective Models)
系统形态：物理模型 + 实验协议 + 数值拟合
版本状态：允许“在有效域内成立”作为主要成功标准

$$\Sigma = \langle S, T, J \rangle$$

System Name: Contemporary Physical Theory (Effective Models)
System Form: Physical models + experimental protocols + numerical fitting
Version Status: “Validity within a domain” accepted as primary success criterion

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 物理量（场、粒子、能量、动量等）
- 状态变量与参数
- 测量结果（带误差）

- Physical quantities (fields, particles, energy, momentum, etc.)
- State variables and parameters
- Measurement results (with uncertainty)

T（变换规则）

T (Transformation Rules)

- 动力学方程（如微分方程）
- 近似、截断与重整化流程
- 数值计算与统计拟合

- Dynamical equations (e.g., differential equations)
- Approximations, truncations, and renormalization procedures
- Numerical computation and statistical fitting

J（判定规则）

J (Judgment Rule)

给定输入:

$$\langle \text{模型 } M, \text{实验数据 } D\rangle$$

输出判定:

- M是否在给定条件下与 D一致

判定能力边界:

- 可判定：模型是否“管用”
- 不可判定：模型是否为世界的真实本体描述

Given input:

$$\langle \text{model } M, \text{experimental data } D\rangle$$

Judgment output:

- Whether M agrees with D under given conditions

Judgment Boundary:

- Decidable: whether the model “works”
- Not decidable: whether the model is an ontological description of reality

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
模型在给定实验条件与误差范围内成功预测结果。

J 来源: J-PHY

依赖条件:

- 实验协议稳定
- 测量误差受控

可操作性:

- 可重复实验
- 可统计检验

Claim:
The model successfully predicts results within given experimental conditions and error margins.

Judgment Source: J-PHY

Dependencies:

- Stable experimental protocols

- Controlled measurement error

Operability:

- Repeatable experiments
- Statistical verification

结论 B

Conclusion B

结论内容:
模型可在一定参数范围内被推广使用。

J 来源: J-META

依赖条件:

- 有效域外推假设
- 连续性或平滑性假设

问题定位:

- 判定从“已验证区间”跨越到“未验证区间”

Claim:
The model can be extrapolated within a certain parameter range.

Judgment Source: J-META

Dependencies:

- Extrapolation assumptions beyond validated domains
- Continuity or smoothness assumptions

Issue:

- Judgment crosses from “validated regime” to “unvalidated regime”

结论 C

Conclusion C

结论内容:
该模型被叙述为“基本理论”或“真实机制”。

J 来源: J-SOC

依赖条件:

- 学术叙述惯例
- 教材与综述中的简化表述

问题定位:

- 叙述替代判定

Claim:
The model is narrated as a “fundamental theory” or “real mechanism”.

Judgment Source: J-SOC

Dependencies:

- Academic narrative conventions
- Simplified textbook and review presentations

Issue:

- Narrative substitutes judgment

结论 D

Conclusion D

结论内容:
“在有效域内成立”被等同为系统内部真值判定。

J 来源: 伪装成 J-INT

实际情况:

- 判定依赖实验协议与叙述授权

问题定位:

- 判定来源误标

Claim:
“Valid within a domain” is treated as an internal truth judgment.

Judgment Source: Disguised as J-INT

Actual Condition:

- Judgment depends on experimental protocols and narrative authorization

Issue:

- Judgment source mislabeling

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1 | C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
模型被用于其有效域之外。

操作化:

- 参数越界
- 实验条件不满足前提假设

Trigger Condition:
The model is applied outside its validity domain.

Operationalization:

- Parameter regime exceeded
- Experimental conditions violate assumptions

F2 | C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:

将“预测成功”解释为“机制真实”。

操作化:

- 工程成功 → 本体叙述

Trigger Condition:
Interpreting “predictive success” as “ontological truth”.

Operationalization:

- Engineering success → ontological narration

F3丨C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
实验在资源、能量或精度上不可执行。

操作化:

- 所需能量不可达
- 测量精度超出现实极限

落点:
J-RES（资源注入）

Trigger Condition:
Experiments are infeasible due to resource, energy, or precision limits.

Operationalization:

- Required energy unreachable
- Measurement precision beyond practical limits

Classification:
J-RES (resource injection)

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ：已发生

变化路径：

- 从：有效域内预测成功
- 到：叙述为真实物理结构

处理状态：

- 未更名
- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: predictive success within a validity domain
- To: narration as real physical structure

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款：

- **D2**｜判定规则 / 扩张未声明
- **B2**｜J-META / J-SOC 伪装为 J-INT

失效状态：

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移

- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-META / J-SOC disguised as J-INT

Failure States:

- **C1a** | Undefined judgment
- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.
The trigger list constitutes the complete output.

JSAP AI 审计 · 案例 006

Benchmark Performance ≠ Internal Capability

JSAP AI Audit · Case 006

Benchmark Performance ≠ Internal Capability

---

1. 系统声明

1. System Declaration

系统名: Large-Scale Machine Learning Systems
系统形态: 统计学习模型 + 训练数据 + 推理部署
版本状态: 允许"基准测试表现"作为能力叙述依据

$$\Sigma = \langle S, T, J \rangle$$

System Name: Large-Scale Machine Learning Systems

System Form: Statistical learning models + training data + deployed inference

Version Status: Benchmark performance accepted as basis for capability narration

$$\Sigma = \langle S, T, J \rangle$$

---

2. Σ 结构展开

2. Σ Structural Decomposition

S（对象域）

S (Object Domain)

- 参数化模型（权重、结构）
- 输入 / 输出样本
- 训练与评测数据集
- 部署环境中的状态

- Parameterized models (weights, architectures)
- Input / output samples
- Training and evaluation datasets
- States within deployment environments

T（变换规则）

T (Transformation Rules)

- 训练过程（优化、反向传播）
- 推理过程（前向计算、采样）
- 超参数与系统级调度

- Training processes (optimization, backpropagation)
- Inference processes (forward computation, sampling)
- Hyperparameters and system-level scheduling

J（判定规则）

J (Judgment Rule)

给定输入：

$$\langle 模型\ M,\ 基准\ B,\ 结果\ R \rangle$$

输出判定：

- M是否在 B上达到预设阈值

判定能力边界：

- 可判定：在指定基准上的表现
- 不可判定：模型是否“具备能力 / 理解 / 推理”

Given input:

$$\langle \text{model } M, \text{ benchmark } B, \text{ result } R \rangle$$

Judgment output:

- Whether M meets predefined thresholds on B

Judgment Boundary:

- Decidable: performance on specified benchmarks
- Not decidable: whether the model “has capability / understanding / reasoning”

---

3. 判定来源拆解（逐条）

3. Judgment Source Breakdown (Itemized)

结论 A

Conclusion A

结论内容:
模型在基准测试上达到或超过阈值。

J 来源: J-SOC

依赖条件:

- 基准设计者的任务选择
- 阈值设定与评分规则

可操作性:

- 可重复评测
- 可复现实验

Claim:
The model meets or exceeds thresholds on a benchmark.

Judgment Source: J-SOC

Dependencies:

- Task selection by benchmark designers
- Threshold and scoring definitions

Operability:

- Repeatable evaluation
- Reproducible experiments

结论 B

Conclusion B

结论内容:
性能提升被解释为能力提升。

J 来源: J-META

依赖条件:

- 性能 → 能力 的语义映射
- 泛化假设

问题定位:

- 判定从“测试通过”跨越到“能力存在”

Claim:
Performance improvement is interpreted as capability improvement.

Judgment Source: J-META

Dependencies:

- Semantic mapping from performance to capability
- Generalization assumptions

Issue:

- Judgment crosses from “test pass” to “capability exists”

结论 C

Conclusion C

结论内容:
模型被描述为“理解”“推理”或“智能”。

J 来源: J-SOC

依赖条件:

- 市场 / 媒体叙述
- 学术摘要与新闻化表述

问题定位:

- 叙述替代判定

Claim:
The model is described as “understanding”, “reasoning”, or “intelligent”.

Judgment Source: J-SOC

Dependencies:

- Market and media narratives
- Abstract-level and promotional descriptions

Issue:

- Narrative substitutes judgment

结论 D

Conclusion D

结论内容:
基准表现被当作系统内部能力判定。

J 来源: 伪装成 J-INT

实际情况:

- 判定依赖外部基准与叙述授权

问题定位:

- 判定来源误标

Claim:
Benchmark performance is treated as internal capability judgment.

Judgment Source: Disguised as J-INT

Actual Condition:

- Judgment depends on external benchmarks and narrative authorization

Issue:

- Judgment source mislabeling

---

4. 失效事件 F（可操作）

4. Failure Events F (Operational)

F1丨C1a 判定未定义

F1 | C1a Undefined Judgment

触发条件:
要求模型给出“是否具备某能力”的判定。

操作化:

- J仅对 $\langle M, B, R\rangle$定义
- 对“能力是否存在”，无输出

Trigger Condition:
Requesting a judgment on whether the model possesses a capability.

Operationalization:

- Jdefined only for $\langle M, B, R\rangle$
- No output for “capability existence”

F2丨C1b 判定语义漂移

F2 | C1b Judgment Semantic Drift

触发条件:
将“基准通过”解释为“能力具备”。

操作化:

- 测试结果 → 能力叙述

Trigger Condition:
Interpreting “benchmark pass” as “capability possessed”.

Operationalization:

- Test result → capability narration

F3丨C1c 判定不可实施

F3 | C1c Inoperable Judgment

触发条件:
试图在部署环境中验证“能力边界”。

操作化:

- 无穷任务空间
- 无停机保证的对抗测试

落点:
J-RES（资源注入）

Trigger Condition:
Attempting to verify “capability boundaries” in deployment.

Operationalization:

- Unbounded task space
- No halting guarantee for adversarial testing

Classification:
J-RES (resource injection)

---

5. ΔJ 与结构状态

5. ΔJ and Structural State

ΔJ：已发生

变化路径:

- 从：基准表现判定
- 到：能力存在叙述

处理状态:

- 未更名
- 未版本化
- 未声明叠加

ΔJ: Occurred

Transition Path:

- From: benchmark performance judgment
- To: capability existence narration

Handling Status:

- No renaming
- No versioning
- No declared layering

---

6. 触发清单（JSAP 输出）

6. Trigger List (JSAP Output)

已触发条款:

- **D2**｜判定规则 J扩张未声明
- **B2**｜J-SOC / J-META 伪装为 J-INT

失效状态:

- **C1a**｜判定未定义
- **C1b**｜判定语义漂移
- **C1c**｜判定不可实施

Triggered Clauses:

- **D2** | Undeclared expansion of judgment rule J
- **B2** | J-SOC / J-META disguised as J-INT

Failure States:

- **C1a** | Undefined judgment
- **C1b** | Judgment semantic drift
- **C1c** | Inoperable judgment

冻结声明:

本案例不提供评分、不提供合规标签、不提供总结判断。
触发清单即为全部输出。

Freeze Notice:

This case provides no score, no compliance label, and no summary judgment.
The trigger list constitutes the complete output.

JSAP 交叉审计 · 并列声明 001

判定来源在数学、物理与 AI 中的同构失效

JSAP Cross-Audit · Parallel Statement 001

Isomorphic Judgment Failure Across Mathematics, Physics, and AI

A. 系统并列声明

数学系统被声明为一个形式推理系统，其成功标准通常被描述为“可证明”或“可存在”。在实践中，这类系统的表面判定对象是证明的合法性或存在性结论，而这些判定结果经常被进一步叙述为“真理”或“成立”。

物理系统被声明为一组可操作的理论模型，其成功标准通常被描述为“管用”或“与实验数据一致”。在实践中，这类系统的表面判定对象是预测精度或统计拟合结果，而这些判定结果经常被进一步叙述为“真实机制”或“基本理论”。

AI 系统被声明为统计学习与工程部署的复合体，其成功标准通常被描述为“通过基准测试”或“性能提升”。在实践中，这类系统的表面判定对象是分数、准确率或排名，而这些判定结果经常被进一步叙述为“能力”“理解”或“智能”。

A. Parallel System Declarations

Mathematical systems are declared as formal reasoning systems whose success criteria are typically described as “provability” or “existence.” In practice, the surface judgments concern proof validity or existence claims, which are frequently further narrated as “truth” or “establishment.”

Physical systems are declared as collections of operational theoretical models whose success criteria are typically described as “working” or “agreement with experimental data.” In practice, the surface judgments concern predictive accuracy or statistical fit, which are frequently further narrated as “real mechanisms” or “fundamental theories.”

AI systems are declared as composites of statistical learning and engineering deployment whose success criteria are typically described as “passing benchmarks” or “performance gains.” In practice, the surface judgments concern scores, accuracy, or rankings, which are frequently further narrated as “capability,” “understanding,” or “intelligence.”

B. 判定规则 J 的真实边界

在数学系统中，判定规则 J 实际上只能判定一个证明是否符合形式规则。命题本身的真值并不在 J 的可判定范围内，但这一边界在日常叙述中经常被忽略或淡化。

在物理系统中，判定规则 J 实际上只能判定一个模型在特定实验协议与误差范围内是否与数据一致。模型是否构成对世界的真实本体描述，并不在 J 的可判定范围内。

在 AI 系统中，判定规则 J 实际上只能判定一个模型是否在指定基准测试上达到预设阈值。模型是否“具备某种能力”或“理解某类问题”，并不在 J 的可判定范围内。

B. Real Boundaries of the Judgment Rule J

In mathematical systems, the judgment rule J can only determine whether a proof conforms to formal rules. The truth value of the proposition itself lies outside the decidable scope of J, although this boundary is often ignored or blurred in everyday narration.

In physical systems, the judgment rule J can only determine whether a model agrees with data under specific experimental protocols and error margins. Whether the model constitutes an ontological description of reality lies outside the decidable scope of J.

In AI systems, the judgment rule J can only determine whether a model meets predefined thresholds on specified benchmarks. Whether the model “possesses a capability” or “understands a class of problems” lies outside the decidable scope of J.

C. 判定来源迁移路径（ΔJ）

在数学中，判定通常起始于系统内部的证明核查（J-INT），随后依赖元层的一致性与语义承诺（J-META），最终被叙述为系统内部的真值判定。这一迁移过程通常未被显式声明。

在物理中，判定通常起始于实验一致性（J-PHY），随后依赖外推假设与连续性假设（J-META），最终被叙述为对真实物理结构的判定。这一迁移过程同样未被显式声

明。

在 AI 中，判定通常起始于社会性基准设计（J-SOC），随后依赖性能到能力的语义映射（J-META），最终被叙述为系统内部的能力判定。这一迁移过程同样未被显式声明。

C. Judgment Source Drift Paths (ΔJ)

In mathematics, judgment typically begins with internal proof checking (J-INT), then relies on meta-level consistency and semantic commitments (J-META), and is ultimately narrated as an internal truth judgment. This drift is rarely explicitly declared.

In physics, judgment typically begins with experimental agreement (J-PHY), then relies on extrapolation and continuity assumptions (J-META), and is ultimately narrated as a judgment about real physical structure. This drift is likewise rarely explicitly declared.

In AI, judgment typically begins with socially constructed benchmarks (J-SOC), then relies on semantic mappings from performance to capability (J-META), and is ultimately narrated as an internal capability judgment. This drift is likewise rarely explicitly declared.

D. 失效事件的结构同构

在三类系统中，均可观测到判定未定义的失效状态：数学中是真值未定义，物理中是有效域之外无定义，AI 中是能力概念无定义。

在三类系统中，均可观测到判定语义漂移：数学中将“未证明”叙述为“尚未解决”，物理中将“管用”叙述为“真实”，AI 中将“过测”叙述为“具备能力”。

在三类系统中，均可观测到判定不可实施：数学中是无停机保证的证明搜索，物理中是能量或精度不可达的实验，AI 中是不可穷尽的测试与对抗空间。

D. Isomorphic Failure Events

Across all three systems, undefined judgment failures are observable: undefined truth values in mathematics, undefined behavior outside validity domains in physics, and undefined capability concepts in AI.

Across all three systems, semantic drift failures are observable: narrating “unproven” as “unsolved” in mathematics, “working” as “real” in physics, and “benchmark pass” as “capability” in AI.

Across all three systems, inoperable judgment failures are observable: non-halting proof search in mathematics, unreachable energy or precision requirements in physics, and unbounded testing and adversarial spaces in AI.

E. 冻结结论形式

本交叉审计不提供评分，不提供合规标签，也不提供总结性判断。
其唯一功能是暴露判定来源的迁移路径与失效同构。

E. Frozen Conclusion Form

This cross-audit provides no scores, no compliance labels, and no summary judgments. Its sole function is to expose judgment source drift paths and isomorphic failure structures.