

## 구조적 닫힘(Structural Closure):

## 반복 실패를 봉인하고 결심 경로를 단일화하는 실행 아키텍처

-LLM 에이전트의 “빈칸(Blank)”과 “결심 분기”를 구조적으로 봉인하는 제약 기반 실행 설계-

### 저자

이충삼

### 구분

독립 연구(비상업)/ Independent Research (Non-Commercial)

### 식별자

ORCID: 0009-0008-8890-3496

### 작성일

2026년 2월 11일

### 공개/등재

OSF 공개 등재 (비상업)

GitHub 공개 등재 (비상업)

### 사용 및 라이선스 고지

본 논문은 **비상업적 연구 및 학술 목적으로만** 공개됩니다.

단순 열람 및 인용을 넘어서는 모든 활용(재배포, 강의자료화, 시스템/모델 적용, 실전 의사결정 적용 등)은

**반드시 사전에 저자와 협의(대화) 후 명시적 허가를** 받아야 합니다.

본 논문의 전부 또는 일부에 대한 **상업적 사용은 엄격히 금지**됩니다.

본 논문은 동일 연구 내용을 담은 **국문/영문 논문을 병렬로 공개한 것**이며,  
두 논문은 언어만 다를 뿐 **동등한 학술적 효력**을 가지며 **동일 조건이 적용**됩니다.

### 연락 및 사전 협의

저자: 이충삼

사전 협의 요청: GitHub Issue

## 초 록

본 연구는 대규모 언어모델 기반 에이전트 및 추론 시스템에서 반복적으로 관찰되는 실패를 “정답 생성의 한계”가 아니라, 결심이 이루어지는 경로 자체가 흔들리는 문제(결심 경로 불안정)로 재정의한다. 기존 접근은 모델 규모 확장 또는 사후 안정화(post-hoc smoothing/patching)<sup>1)</sup>를 통해 결과를 보정하는 방식으로 대응해 왔으나, 이러한 방식은 결심의 권한 경계를 흐리게 하여 조기 확정, 사후 수정, 근거 공백, 규칙 증식의 악순환을 유발한다. 이에 본 논문은 결심 경로를 단일화하고 공백을 봉인함으로써, 운용 중에도 결심 규격이 붕괴하지 않도록 하는 “구조적 닫힘(Structural Closure)” 실행 구조를 제안한다.

제안 구조는 NCAF(상위 제약/정렬 프레임), PCM(결심 허용·격리·중단을 강제하는 절차 레일), PVM(허용 이후 제한적으로 판례를 참조하되 규칙 비증식성을 유지하는 메커니즘)으로 구성된다. PCM은 누락·불일치·순서 위반과 같은 무결성 위반을 탐지하면 결론 산출을 즉시 중단하는 Fail-Stop<sup>2)</sup>을 수행하며, 결심이 커밋된 이후에는 어떤 모듈도 결론을 수정할 수 없는 출력 불변(Output Immutability)<sup>3)</sup>을 강제한다. PVM은 PCM이 결심을 허용한 이후에만 작동하며, 결론을 변경하지 않고 경고/위험도/예측 범위 조정과 같은 메타 수준의 신호만을 제공한다. 이로써 결심 과정은 “조기 확정의 유혹”과 “사후 수정의 도피”로부터 분리되어, 실행 중에도 동일한 절차 규격과 권한 경계가 유지된다.

실증은 성능 비교가 아니라 운용 무결성의 관점에서 수행된다. 본 연구는 실제 운용 로그에서 생성된 온톨로지 좌표(onto\_tmp-5.csv)를 기반으로, 결심 레일의 분기(허용/격리/중단/경고)가 어떻게 기록·재현되는지를 합성 TRACE 예시로 제시한다. 이 TRACE는 입력 원시값의 재현이 아니라, PCM-IC-GroupHat-FINAL-PVM으로 이어지는 절차 재현을 목적으로 하며, 결심 커밋 이후 결론이 수정되지 않는다는 불변성, 그리고 무결성 위반 시 결심이 금지되는 Fail-Stop의 정상 동작을 확인한다.

본 연구는 모델 성능 우위(benchmark superiority)를 주장하지 않으며, 통계적 일반화를 목표로 하지 않는다. 또한 원시값 전면 공개 대신, 절차 규격과 TRACE 기반 재현 단위를 제공한다. 본 논문이 제시하는 구조적 닫힘은 에이전트의 불안정성과 반복 실패를 “결과

1) 본 논문에서 ‘사후 안정화’란 결심(최종 출력) 이후 결과를 매끈하게 만들거나 일관되게 보이도록 값/판정을 재가공·재조정하는 모든 후처리(스무딩, 패치, 임시 규칙 덧댐)를 의미한다. 본 논문은 이러한 방식이 실패를 ‘개선’이 아니라 ‘은폐’로 바꾸는 지점을 문제로 삼으며, 결심 이후 수정이 구조적으로 불가능하도록 절차를 설계한다.

2) Fail-Stop은 무결성 위반(필수 입력 누락, 불일치, 순서 위반 등)이 탐지되면 ‘부정확한 결론을 내지 않는 것’을 정상 동작으로 간주하고, 결심 산출을 즉시 중단하는 설계 원리이다. 본 논문에서는 결심 경로가 흔들릴 때 “일단 출력”을 금지하기 위한 안전 레일로 사용된다.

3) ‘출력 불변’은 결심이 커밋된 이후 어떤 모듈도 결론(방향/판정/최종 출력)을 수정할 수 없다는 규정이다. 이는 ‘사후 보정’이 구조적으로 침투하는 것을 차단하기 위한 핵심 불변 조건이며, TRACE는 이 불변 조건이 지켜졌음을 재현 가능하게 기록한다.

보정"이 아닌 "결심 경로 규격화"로 해소하는 실행 아키텍처로서, 향후 신뢰 가능한 추론 시스템의 설계·운용·기록 체계에 적용 가능한 기준점을 제공한다.

핵심어: 구조적 단힘, 결심 경로, 공백 봉인, NCAF, PCM, PVM, Fail-Stop, 출력 불변, TRACE, 온톨로지 로그

**본 연구는 비상업적 연구 및 학술 목적 공개를 전제로 한다.**

**본 자료의 전부 또는 일부를 시스템에 적용하거나 실전 의사결정에 사용하는 행위는 사전 협의 및 명시적 허가 없이는 허용되지 않는다.**

# 목 차

## 제 I 장 서론

### 1.1 연구 목적과 문제 정의

- 1.1.1 연구 목적: 결심 경로 안정화
- 1.1.2 문제 정의: 결심 경로 오염
- 1.1.3 연구질문: 공백과 혼선의 봉인

### 1.2 연구 가설: 구조적 닫힘

- 1.2.1 공백 봉인
- 1.2.2 결심 경로 단일화
- 1.2.3 출력 불변과 TRACE

### 1.3 연구 범위와 실증 구성

- 1.3.1 실증 1: 방향 판단(설계·연산)
- 1.3.2 실증 2: 진동폭 처리(운용)
- 1.3.3 본 편에서 주장하지 않는 범위

### 1.4 기여

- 1.4.1 NCAF·PCM·PVM 계층 구현 명세
- 1.4.2 규칙 비증식성과 판례 추적 원리
- 1.4.3 예측 시스템 운

## 제 II 장 LLM 실패의 오해된 원인과 결심 경로 문제의 정식화

### 2.1 실패는 줄지 않았다, 다만 형태만 바뀌었다

- 2.1.1 그럴듯한 오답의 확산
- 2.1.2 실패의 은폐와 발견 불가능성
- 2.1.3 결심 시점 불명확성

### 2.2 “성능 문제” 프레임의 실패

- 2.2.1 성능 향상과 실패 반복의 공존
- 2.2.2 그럴듯함이 만드는 위험

### 2.3 “환각”이라는 잘못된 표적

- 2.3.1 환각은 원인이 아니라 결과
- 2.3.2 승격 경로가 열려 있을 때의 위험

## **2.4 규칙 증식은 해결이 아니라 은폐다**

- 2.4.1 예외의 예외 구조
- 2.4.2 경계 붕괴와 충돌
- 2.4.3 실패의 삭제와 기록 오염

## **2.5 KG 및 스코프 제한의 한계**

- 2.5.1 스코프 제한의 장점
- 2.5.2 결심 오염과의 분리 실패
- 2.5.3 결심 경로 봉인의 필요성

## **2.6 사후 안정화라는 위험한 관행**

- 2.6.1 보정이 예측을 연출로 바꾸는 지점
- 2.6.2 재현성 붕괴의 발생 조건

## **2.7 결심 개념의 부재**

- 2.7.1 결심 미정의가 만드는 혼선
- 2.7.2 결심 레이어 단일화 필요성

## **2.8 소결: 결심 경로 문제의 재정의**

- 2.8.1 결심 허용 조건의 필요성
- 2.8.2 절차 레일 강제의 필요성
- 2.8.3 구조적 단힘으로의 전환

## **제3장 구조적 단힘 아키텍처 개요**

### **3.1 설계 목표와 금지 규정**

- 3.1.1 조기결정 금지
- 3.1.2 사후변경 금지
- 3.1.3 결심 레이어 단일화

### **3.2 레이어 분리 원칙(NCAF 개요)**

- 3.2.1 계층 레이어의 역할 고정
- 3.2.2 라벨 레이어의 역할 고정

- 3.2.3 결심 레이어의 유일성
- 3.2.4 레이어 침범 금지 규정

### **3.3 결심 허용 규정(PCM 개요)**

- 3.3.1 결심 허용의 최소 조건
- 3.3.2 절차 레일(순서 강제)
- 3.3.3 중단 규정(누락/불일치/순서 위반)

### **3.4 TRACE와 책임성**

- 3.4.1 출력 불변 기록
  - 3.4.2 재현성 최소 조건
  - 3.4.3 운영 로그 구조화
- 3.5 장 소결

## **제IV장 코어 기술 명세: NCAF-PCM-PVM의 역할 분리와 통합 실행 구조**

### **4.1 장 개요와 명세의 목적**

- 4.1.1 명세 대상의 한정
- 4.1.2 명세의 논리적 지위

### **4.2 NCAF: 인지 정렬 프레임과 레이어 권한 체계**

- 4.2.1 NCAF의 정의
- 4.2.2 레이어별 권한과 금지 행위
- 4.2.3 레이어 분리의 귀결: 책임성과 재현성

### **4.3 PCM: 결심 허용 제약(Principled Constraint Model)**

- 4.3.1 PCM의 정의: 결심은 허용되는 전이다
- 4.3.2 결심 허용 최소 조건(Minimum Admissibility)
- 4.3.3 절차 레일 강제(Sequence Lock)
- 4.3.4 PCM의 산출물: 결심 허용/격리/중단

### **4.4 PVM: 판례 참조(Past-Case Validation Mechanism)**

- 4.4.1 PVM의 정의: 규칙이 아니라 판례
- 4.4.2 작동 시점: PCM 이후
- 4.4.3 판례 축적과 실패의 자산화

## **4.5 통합 실행 구조: 결심 경로 봉인의 완결**

### 4.5.1 통합 파이프라인

### 4.5.2 오염 차단 지점

### 4.5.3 실증과의 연결

## **4.6 장 소결**

## **제V장 실증 1: 방향 판단 프로세서의 설계·연산 차원 구조적 닫힘**

## **5.1 문제 설정과 I/O 규격**

### 5.1.1 입력 시퀀스와 $\Delta t$ 규정

### 5.1.2 출력 정의(UP/DOWN)

## **5.2 레이어 분리: 계측·라벨과 결심의 분리**

### 5.2.1 기준1~4의 결정 금지

### 5.2.2 기준5의 유일 결심권

### 5.2.3 누락·불일치 중단 규정

## **5.3 결심 경로 단일화**

### 5.3.1 BASE와 GroupHat의 분리

### 5.3.2 출력 절차 레일 고정

## **5.4 결심 허용 최소 조건(PCM 적용)**

### 5.4.1 REGIME-LOCK

### 5.4.2 GATE\_PRE

### 5.4.3 IC

### 5.4.4 GroupHat

### 5.4.5 FINAL

### 5.4.6 요약

## **5.5 패치·예외의 위치 고정**

### 5.5.1 패치 허용 범위(결심층 한정)

### 5.5.2 규칙 비증식성 유지

## **5.6 장 소결**

## **제VI장 실증 2: 진동폭 처리 프로세서의 운용 차원 구조적 닫힘**

## **6.1 진동폭 처리 문제와 구조적 닫힘의 필요성**

- 6.1.1 수치 예측 영역에서의 결심 오염 위험
- 6.1.2 조기확정과 사후안정화의 결합
- 6.1.3 운용 차원 달함의 목표

## **6.2 진동폭 처리 파이프라인과 결심 경로 봉인**

- 6.2.1 파이프라인 개요
- 6.2.2 Mandatory Isolation과 IC Filter

## **6.3 실증 데이터의 지위와 한계**

- 6.3.1 174 사례의 성격
- 6.3.2 공개 범위와 해석 제한
- 6.3.3 “하지 않은 주장”의 명시

## **6.4 히트와 미스의 역할 분리**

- 6.4.1 히트: 공통사항의 규칙화
- 6.4.2 미스: 판례로서의 실패
- 6.4.3 규칙 비증식성의 유지

## **6.5 표 기반 운용 실증**

- 6.5.1 표 구성 원칙
- 6.5.2 조기결정 차단 관측
- 6.5.3 사후안정화 차단 관측
- 6.5.4 판례 축적 관측
- 6.5.5 규칙 비증식성 관측

## **6.6 장 소결**

## **제Ⅶ장 논의: 예상 반론에 대한 구조적 봉인**

### **7.1 논의의 기준과 범위**

- 7.1.1 본 장의 기준: 반론 대응이 아니라 전환의 증명
- 7.1.2 논의 범위: 성능 대신 ‘운용 가능한 책임’을 다룬다

### **7.2 반론 1: “구조적 달함은 개념적 말장난이다”**

- 7.2.1 말이 아니라 ‘달히는 것’이 무엇인지 먼저 본다



### **7.3 반론 2: “단일 규칙이 아니라 더 많은 규칙이 필요하다”**

7.3.1 이 반론은 ‘결과 중심’ 사고의 전형이다

7.3.2 두 번째 전환: ‘사후 보정’이 아니라 ‘실패를 남기는 개선’으로 이동한다

### **7.4 반론 3: “KG 스코프 제한이면 충분하다”**

7.4.1 KG는 ‘무엇을 아는가’를 줄이지만 ‘어떻게 결심하는가’를 고정하지 못한다

7.4.2 이 반론은 ‘지식 통제’로 ‘책임’을 대체하려는 시도다

### **7.5 반론 4: “중간층·상위층 구현은 과도하다”**

7.5.1 단일 층 해법은 구현이 아니라 책임을 무너뜨린다

7.5.2 절차의 경제는 계층 분리를 요구한다

### **7.6 반론 5: “PVM은 사후 보정과 다르지 않다”**

7.6.1 PVM이 작동하는 순간은 ‘결심 이후’로 고정된다

7.6.2 세 번째 전환: 재현을 ‘숫자’가 아니라 ‘절차’로 정의한다

## **7.7 장 소결**

## **제Ⅷ장 결론**

### **8.1 연구 요약**

8.1.1 문제의 재정의

8.1.2 해결의 방향: 결심 경로 봉인

### **8.2 핵심 기여**

8.2.1 구조적 닫힘의 실행 규격화

8.2.2 NCAF·PCM·PVM의 역할 분리와 통합 구조 제시

8.2.3 실증 1: 방향 판단 프로세서의 설계·연산 차원 닫힘 제시

8.2.4 실증 2: 진동폭 처리 프로세서의 운용 차원 닫힘 입증

### **8.3 해석과 함의**

8.3.1 “정확도”보다 “책임성과 재현성”의 우선

8.3.2 규칙 증식의 유혹과 규칙 비증식성의 필요

8.3.3 KG 스코프 제한과의 관계

### **8.4 한계**

8.4.1 실증 범위의 제한

#### 8.4.2 판례 활용의 운영 설계 과제

### 8.5 향후 연구

#### 8.5.1 결심 경로 봉인의 일반화

#### 8.5.2 실패의 표준 좌표화와 판례 상호운용

#### 8.5.3 결심 책임의 검증 가능성

### 8.6 결론

### 참고문헌

### 부록

#### 부록 A 용어 및 약어

#### 부록 B 실행 규격(TRACE)

#### 부록 C NCAF 규정

#### 부록 D PCM 규정

#### 부록 E PVM 규정

#### 부록 F 그림·표 목록

#### 부록 G. 합성 TRACE 3케이스(절차 재현용)

## 제 I 장 서론

### 1.1 연구 목적과 문제 정의

#### 1.1.1 연구 목적: 결심 경로 안정화

LLM 기반 예측 및 의사결정 시스템이 실무로 들어오면서, 시스템이 “무엇을 아는가”보다 “어떻게 결론에 도달하는가”가 더 큰 문제로 부상했다. 예측 시스템은 결과의 유창함이나 설명 가능성이 아니라, 동일 조건에서의 재현성과 책임성, 그리고 실패가 남겨지는 방식에 의해 평가되어야 한다. 그러나 실제 운영 환경에서 실패는 단일한 오답으로 나타나지 않는다. 결론의 근거가 약한 상태에서 결론이 먼저 확정되거나, 산출된 결과가 사후적으로 안정화(보정)되어 기록이 변형되거나, 설득력 있는 연결이 검증 없이 결론으로 승격되는 형태로 누적된다. 이 누적은 데이터가 부족해서라기보다 결론이 생성되는 과정에 구조적 제약이 부족해서 발생한다.

본 연구의 목적은 예측 시스템의 실패를 “추론 성능”의 문제로 환원하지 않고, 결심이 발생하는 경로 자체를 안정화하는 실행 구조를 제시하는 데 있다. 다시 말해, 결심이 언제, 어디서, 어떤 조건과 절차로 허용되는지를 명시적으로 고정함으로써, 조기결정과 사후안정화, 그리고 근거 없는 결론 승격이 발생하는 경로를 차단한다. 이를 통해 본 연구는 예측 시스템을 “결과를 그럴듯하게 만드는 장치”가 아니라 “결심의 발생 조건을 통제하는 장치”로 재정의한다.[1],[2]

#### 1.1.2 문제 정의: 결심 경로 오염

본 연구가 다루는 핵심 문제는 결심 경로 오염이다. 결심 경로 오염이란, 예측·상태·패턴과 같은 중간 산출물이 절차적 제약 없이 결론 단계로 유입되어, 결심이 발생하는 지점이 흐려지고 결론 승격의 기준이 불명확해지는 현상을 의미한다. 이 오염은 다음과 같은 경로를 통해 실무적으로 나타난다.[3],[14]

첫째, 결심 시점이 정의되지 않은 상태에서 결론이 ‘어느 순간’ 자연스럽게 확정된다. 운영자는 결론이 확정된 이유를 사후적으로 설명할 수 있을지 몰라도, 결론이 확정되어도 되는 최소 조건이 무엇이었는지, 그 조건이 충족되었는지는 확인할 수 없다.

둘째, 결론이 확정된 이후 결과가 안정화(보정)되는 순간, 실패는 실패로 남지 못하고 기록에서 삭제된다. 이때 시스템은 개선되는 것이 아니라 실패를 가리는 방향으로 진화한다. 결과가 더 “좋아 보이기” 시작하면, 오염은 더 발견하기 어렵게 된다.

셋째, 그럴듯한 연결이 검증 없이 결론으로 승격되는 순간, 시스템은 지식을 확장하는 것이

아니라 결론을 합성하는 장치가 된다. 설득력은 증가하나 책임성과 재현성은 감소한다. 예측 시스템에서 가장 위험한 지점은 “틀렸는데 그럴듯한” 결과가 “맞는 것처럼” 운용되는 상황이다.

이러한 현상은 개별 규칙을 더 정교하게 만들거나, 모델 성능을 높이거나, 일부 도메인 지식을 보강하는 방식으로 근본적으로 해소되지 않는다. 왜냐하면 문제의 중심은 결론의 내용이 아니라 결론의 생성 경로이기 때문이다.

### 1.1.3 연구 질문: 공백과 혼선의 봉인

모든 시스템은 공백을 가진다. 예측 시스템은 특히 그렇다. 모든 상태를 완전하게 정의할 수 없고, 모든 변수를 사전에 포함할 수 없으며, 모든 예외를 규칙으로 흡수할 수도 없다. 따라서 문제는 공백의 존재 자체가 아니라, 공백이 결론 단계로 흘러 들어가 결론을 오염시키는 경로가 열려 있다는 점이다.[1],[3],[14]

본 연구의 질문은 다음으로 정식화된다. 공백과 혼선이 존재하는 환경에서, 결심이 발생할 수 있는 경로를 구조적으로 봉인할 수 있는가. 봉인이란 공백을 제거하는 것이 아니라, 공백이 결론으로 승격되는 절차를 막는 것이다. 이 질문은 기술적 기법의 선택 이전에 실행 구조의 설계 문제로 다루어져야 한다.

## 1.2 연구 가설: 구조적 닫힘

### 1.2.1 공백 봉인

구조적 닫힘의 첫 번째 가설은 공백 봉인이다. 공백 봉인은 “모르는 것을 채우지 않는다<sup>4)</sup>”는 선언이 아니라, “모르는 것이 결심을 유발하지 못한다”는 실행 규정이다. 즉, 미정의 상태나 불확실성이 존재하더라도 그것이 결론 단계로 승격되는 경로를 차단한다. 공백이 존재하는 것은 자연스럽고 불가피하지만, 공백을 근거로 결론을 확정하는 것은 운영 실패다. 구조적 닫힘은 이 경계를 절차적으로 고정한다.[14]

공백 봉인은 시스템의 겸손을 강제한다. 그러나 여기서 말하는 겸손은 태도의 문제가 아니라 구조의 문제다. 결론으로 갈 수 있는 경로가 봉인되어 있다면, 시스템은 공백을 공백으로 남기며, 공백을 메우기 위한 무단 추론을 결론으로 승격시키지 않는다. 공백은 삭제되지 않고 남으며, 남아 있는 공백은 이후 운용에서 판례로 축적될 수 있는 관측 대상으로 바뀐다.

---

4) 여기서 ‘공백’은 (1) 관측 불가/미정의 상태, (2) 증거 누락(NA), (3) 상충 증거로 인해 단일 결론을 허용할 수 없는 상태를 포함한다. 공백 봉인은 공백을 제거하거나 임의로 채우는 것이 아니라, 공백이 결심으로 ‘승격’되는 절차를 차단하는 규정이다.

### 1.2.2 결심 경로 단일화

두 번째 가설은 결심 경로 단일화이다<sup>5)</sup>. 결심이 발생할 수 있는 지점이 여러 곳에 분산되어 있거나, 결심으로 가는 절차가 다수로 존재하는 순간, 시스템은 동일 입력에서 상이한 결론을 산출할 구조적 가능성을 갖는다. 이는 모델의 확률적 특성보다 더 근본적인 불안정성의 원천이다.

구조적 단합은 결심이 발생하는 레이어를 단일화하고, 결심이 허용되는 절차 레일을 고정한다. 단일화의 핵심은 “결심은 언제든지 일어나는 이벤트가 아니다”라는 점이다. 결심은 오직 특정 조건을 통과했을 때만 허용되는 전이이며, 그 전이는 규정된 순서로만 발생한다. 절차 레일이 고정되면, 결심 이전 단계가 결론을 침범하는 경로가 차단되고, 결심이 ‘슬며시’ 발생하는 혼선을 제거할 수 있다.[7],[8]

### 1.2.3 출력 불변과 TRACE

세 번째 가설은 출력 불변과 TRACE이다. 출력 불변이란, 결심이 발생한 이후 결과를 수정하지 않는다는 규정이다. 이는 결과를 고집하기 위한 규정이 아니라, 실패를 실패로 남기기 위한 규정이다. 결과가 틀렸다면 틀린 채로 남아야 원인을 분석할 수 있고, 동일 조건에서의 재현성을 확보할 수 있다. 사후 안정화(보정)는 결과를 개선하는 것처럼 보이지만, 실패를 기록에서 삭제함으로써 예측 시스템을 연출 시스템으로 변질시킨다.

TRACE는 출력 불변을 가능하게 하는 운영 장치다. 결심 이전에 어떤 계측과 라벨이 생성되었고, 어떤 절차를 통과해 결심이 허용되었는지를 기록한다. 중요한 점은 TRACE가 “설명”이 아니라 “불변의 실행 기록”으로 기능해야 한다는 것이다. 시스템이 실패했을 때, TRACE는 실패의 책임 소재를 외부로 돌리는 도구가 아니라, 실패를 자산으로 축적하기 위한 관측 좌표가 된다.[4],[5]

## 1.3 연구 범위와 실증 구성

### 1.3.1 실증 1: 방향 판단(설계·연산)

본 논문의 첫 번째 실증은 방향 판단 프로세서를 대상으로 한다. 이 실증의 목적은 성능 비교나 데이터 기반 우열을 주장하는 것이 아니라, 설계·연산 차원에서 결심 경로가 구조적으로 닫혀 있음을 명세로 제시하는 데 있다. 방향 판단은 결심이 발생하는 유일 레이어, 결심 허용 최소 조건, 출력 절차의 순서 고정, 누락·불일치 시 중단 규정 등을 통해 결심 경로 누수를 차단한다. 즉, “결심이 허용되는 구조” 자체를 닫는 것을 입증한다.[2]

5) ‘단일화’는 계산/라벨/보조 모듈이 결론을 직접 산출하거나 수정할 권한을 갖지 못한다는 뜻이다. 결심(최종 방향/출력)은 단일 레이어(결심층)에서만 발생하며, 나머지 레이어는 계측·라벨·검사·격리 신호만 제공한다.

이 장치가 의미하는 바는 단순하다. 방향 판단은 잘 맞히는 기술이기 이전에, 결심이 발생하는 방식이 일관되도록 강제하는 실행 구조다. 이 구조가 닫혀 있어야만, 이후 운용에서 실패를 실패로 남기고 축적할 수 있다.

### 1.3.2 실증 2: 진동폭 처리(운용)

두 번째 실증은 진동폭 처리 프로세서를 대상으로 한다. 진동폭은 예측 결과가 수치로 직접 노출되는 영역이므로, 조기결정과 사후안정화의 유혹이 가장 강하다. “조금만 보정하면 된다”는 관행이 침투하기 쉬운 영역이며, 그 결과 실패가 은폐되기 쉬운 영역이다.[2],[14]

본 연구는 운용 사례(174)를 근거로, 구조적 닫힘이 운용 과정에서도 유지되는지를 제시한다. 핵심은 규칙을 늘려 불규칙을 흡수하는 것이 아니라, 공통사항은 규칙으로 처리하되 불규칙은 실패로 보존하여 판례로 축적하는 규칙 비증식성 원리가 실제 운용에서 어떻게 작동하는지 보여주는 것이다. 이 실증은 표와 구조도를 근거로 제시된다.

### 1.3.3 본 편에서 주장하지 않는 범위

본 논문은 LLM 모델 간 성능 비교, 파라미터 규모의 우열, 학습 기법의 우수성을 주장하지 않는다<sup>6)</sup>. 또한 방향 판단 영역에 대해서는 대규모 운용 데이터에 기반한 “성능 실증”을 제시하지 않는다. 본 편의 핵심은 성능이 아니라 실행 구조이며, 결심 경로의 봉인과 기록 불변성, 그리고 실패의 판례화가 설계와 운용에서 어떻게 구현되는지를 명세와 운용 근거로 제시하는 데 있다.

## 1.4 기여

### 1.4.1 NCAF·PCM·PVM 계층 구현 명세

본 연구는 결심 경로 오염을 차단하기 위해 NCAF·PCM·PVM의 계층적 역할을 분리하고, 각 계층이 무엇을 할 수 있고 무엇을 할 수 없는지를 실행 규정으로 고정한다. 특히 계층·라벨 단계가 결론을 침범하지 못하도록 경계를 설정하고, 결심은 오직 허용된 절차를 통과했을 때만 발생하도록 단일 레일 절차를 강제한다.[1],[2]

### 1.4.2 규칙 비증식성과 판례 축적 원리

본 연구는 실패를 규칙으로 흡수하여 통제하는 접근을 배제한다. 공통사항은 규칙으로 처리하되, 불규칙은 실패로 보존하여 판례로 축적한다. 이 원리는 시스템의 복잡도를 무한히 증가시키지 않으면서도 예측력을 향상시키기 위한 운영 원리로 제시된다. 실패는 제거 대상이 아니라 원인 규명의 자산이며, 예측 시스템은 실패가 남아야 성장할 수 있다.[6],[14]

6) 본 논문의 실증은 벤치마크 우열이나 통계적 일반화를 주장하기 위한 것이 아니라, (1) 결심 레일이 운용 중 붕괴하지 않는지, (2) 조기결정/사후수정이 구조적으로 차단되는지, (3) 실패가 삭제되지 않고 판례로 축적되는지를 확인하기 위한 절차 실증이다.

### 1.4.3 예측 시스템 운영 규율

본 연구는 예측 시스템을 사후 분석 도구로 취급하지 않는다. 결심은 실전 1회 결정을 위한 것으로만 다루며, 출력은 불변으로 기록된다. 사후 안정화를 통한 결과 조정은 금지되며, 운용 로그는 판례 축적을 위한 입력으로 사용된다. 이 운영 규율은 구조적 단함이 단지 설계 논리가 아니라 실제 운용을 건디는 실행 규율임을 보여준다.

## 1.5 논문 구성

본 논문은 다음과 같이 구성된다. 제2장은 LLM 운용 실패의 오해된 원인들을 정리하고, 결심 경로 문제로의 재정의를 통해 구조적 단함의 필요성을 제시한다. 제3장은 구조적 단함 아키텍처를 개요 수준에서 설명하고, 레이어 분리와 절차 강제, TRACE와 책임성의 연결을 정식화한다. 제4장은 NCAF·PCM·PVM을 코어 기술로서 명세하고, 통합 실행 구조에서의 경계 조건과 오염 차단 지점을 제시한다. 제5장은 방향 판단 프로세서의 설계·연산 차원 실증을 통해 결심 경로 단일화와 절차 레일 고정이 어떻게 구현되는지 명세로 제시한다. 제6장은 진동폭 처리 프로세서의 운용 근거를 통해 구조적 단함이 운용 과정에서 유지되는지를 표와 구조도로 제시한다. 제7장은 예상되는 반론을 구조적으로 검토하고, 제8장은 결론과 향후 과제를 제시한다.

## 제2장 LLM 실패의 오해된 원인과 결심 경로 문제의 정식화

### 2.1 실패는 줄지 않았다, 다만 형태만 바뀌었다

#### 2.1.1 그럴듯한 오답의 확산

LLM 기반 시스템의 실패는 더 이상 “즉시 드러나는 오류”의 형태로만 관측되지 않는다. 최근의 실패는 설명 가능하고 문장적으로 정합적인 형태로 산출되며, 외형상 합리적인 근거를 동반한다. 이 점이 문제를 단순 성능 문제로 오인하게 만드는 핵심 요인이다. 과거에는 오류가 눈에 띄었기 때문에 실패가 기록되고 수정될 여지가 있었다. 반면 현재는 오류가 결론의 형태로 매끈하게 포장되어, 운영자가 실패를 실패로 인식하지 못한 채 운용 의사 결정에 반영하는 위험이 커졌다.[14]

이 변화는 단순히 “모델이 좋아졌다”는 관찰로 환원될 수 없다. 그럴듯함의 증가는 ‘오류의 발견 비용’을 증가시키고, 검증의 부담을 사용자의 책임으로 전가한다. 특히 예측 시스템에서 그럴듯한 오답은 단순 오답이 아니라, 의사결정 비용을 직접 증가시키는 위험 요인이다. 즉, 실패의 형태가 바뀌었다는 진술은 수사적 표현이 아니라, 실패가 운영 위험으로 전환되었음을 의미한다.

#### 2.1.2 실패의 은폐와 발견 불가능성

실무 환경에서 실패는 두 단계로 은폐된다. 첫째, 실패가 “그럴듯함” 때문에 발견되지 않는다. 둘째, 설령 실패가 발견되더라도 사후 안정화(보정) 관행에 의해 실패의 원인이 로그에서 삭제된다. 이 두 단계가 결합될 때 실패는 재현 불가능한 사건으로 남는다. 이 상태에서 시스템은 실패로부터 학습하는 것이 아니라, 실패를 재발생 가능한 형태로 보존한다.[14]

여기서 중요한 것은 실패 은폐가 의도적 조작만을 의미하지 않는다는 점이다. 오히려 실무에서는 “결과를 맞추기 위한 선의의 보정”이 실패 은폐를 촉진한다. 실패가 삭제되면, 남는 것은 안정화된 결과뿐이며, 그 결과는 시스템이 개선된 것처럼 보이게 만든다. 그러나 이 개선은 성능 향상이 아니라 기록 오염의 결과다. 예측 시스템에서 기록 오염은 단순 로그 품질 문제가 아니라, 책임성과 재현성을 훼손하는 구조적 결함이다.

#### 2.1.3 결심 시점 불명확성

이러한 실패 은폐의 중심에는 결심 시점의 불명확성이 있다. 많은 시스템은 추론의 “내용”을 설명할 수는 있지만, 결론이 “언제” 확정되었는지, 그 확정이 허용되는 최소 조건이 무엇인지에 대해서는 명시적 규정을 갖지 않는다. 그 결과 결심은 하나의 절차적 사건이 아니라, 출력 생성 과정 어딘가에서 “슬며시 발생하는 상태”가 된다.[14]



결심 시점이 불명확하면 세 가지 문제가 동시에 발생한다. (1) 결론이 조기에 확정되어도 이를 위반으로 판정할 수 없고, (2) 결론 이후의 보정을 위반으로 판정할 수 없으며, (3) 중간 산출물이 결론을 침범하는 경계를 설정할 수 없다. 따라서 결심 시점 불명확성은 성능이나 데이터 이전의 “구조 문제”로 간주되어야 한다. 본 연구는 이 지점에서 실패를 ‘추론 오류’가 아니라 ‘결심 경로 오염’으로 재정의한다.

## 2.2 “성능 문제” 프레임의 실패

### 2.2.1 성능 향상과 실패 반복의 공존

성능 향상은 일반적으로 문제 해결의 기본 처방으로 간주된다. 그러나 LLM 기반 예측 시스템의 실패는 성능 향상 이후에도 반복되는 양상을 보인다. 이는 실패의 상당 부분이 “틀린 답을 생성했다”는 사실 자체보다, “틀린 답이 결론으로 확정되었다”는 절차적 사건에 의해 발생하기 때문이다. 성능이 향상되어도 결심 경로가 열려 있다면, 결론으로 승격되는 조건과 절차는 여전히 불안정하다.[14]

또한 실무 실패는 단일 오답이 아니라, 동일 입력에서 상이 결론이 발생하거나, 같은 결론이 다른 근거로 정당화되는 형태로 나타난다. 이 현상은 모델 정확도의 문제라기보다, 결심이 발생하는 구조가 단일하지 않다는 징후다. 즉 성능은 필요조건일 수 있으나, 결심 경로 안정화의 충분조건이 아니다. 성능을 올릴수록 오히려 결심 경로 오염의 탐지가 어려워지는 역설이 발생할 수 있다.

### 2.2.2 그럴듯함이 만드는 위험

성능 논의가 실패하는 지점은 “그럴듯함”과 “정확성”을 혼동할 때 발생한다. LLM은 언어적 일관성과 설명의 정합성을 생산하는 능력이 크다. 그러나 예측 시스템에서 중요한 것은 설명이 아니라 결심의 정당화 가능성과 재현성이다. 그럴듯함은 종종 검증되지 않은 연결을 근거처럼 보이게 만들고, 운영자가 검증을 생략하도록 유도한다.[3],[14]

더 큰 문제는 그럴듯함이 높은 출력이 실패했을 때 발생한다. 그 경우 운영자는 실패의 원인을 모델이 아니라 사용자 입력, 상황 변수, 혹은 외부 요인으로 돌리기 쉽다. 즉 그럴듯함은 실패 책임의 소재를 흐리게 만든다. 이처럼 성능을 높이는 접근만으로는 책임성과 재현성을 확보할 수 없으며, 결심 경로를 구조적으로 고정하는 별도의 설계가 요구된다.

## 2.3 “환각”이라는 잘못된 표적

### 2.3.1 환각은 원인이 아니라 결과

환각(hallucination)은 흔히 LLM 실패의 핵심 원인으로 지목된다. 그러나 엄밀히 말해 환각은 원인이라기보다 “검증되지 않은 중간 산출물”의 한 형태다. 모든 추론 체계는 불완전

한 가정이나 임시 연결을 생성할 수 있으며, 그것 자체가 즉시 실패를 의미하지는 않는다. 실패가 되는 순간은 그 산출물이 결론으로 확정될 때다.[6],[14]

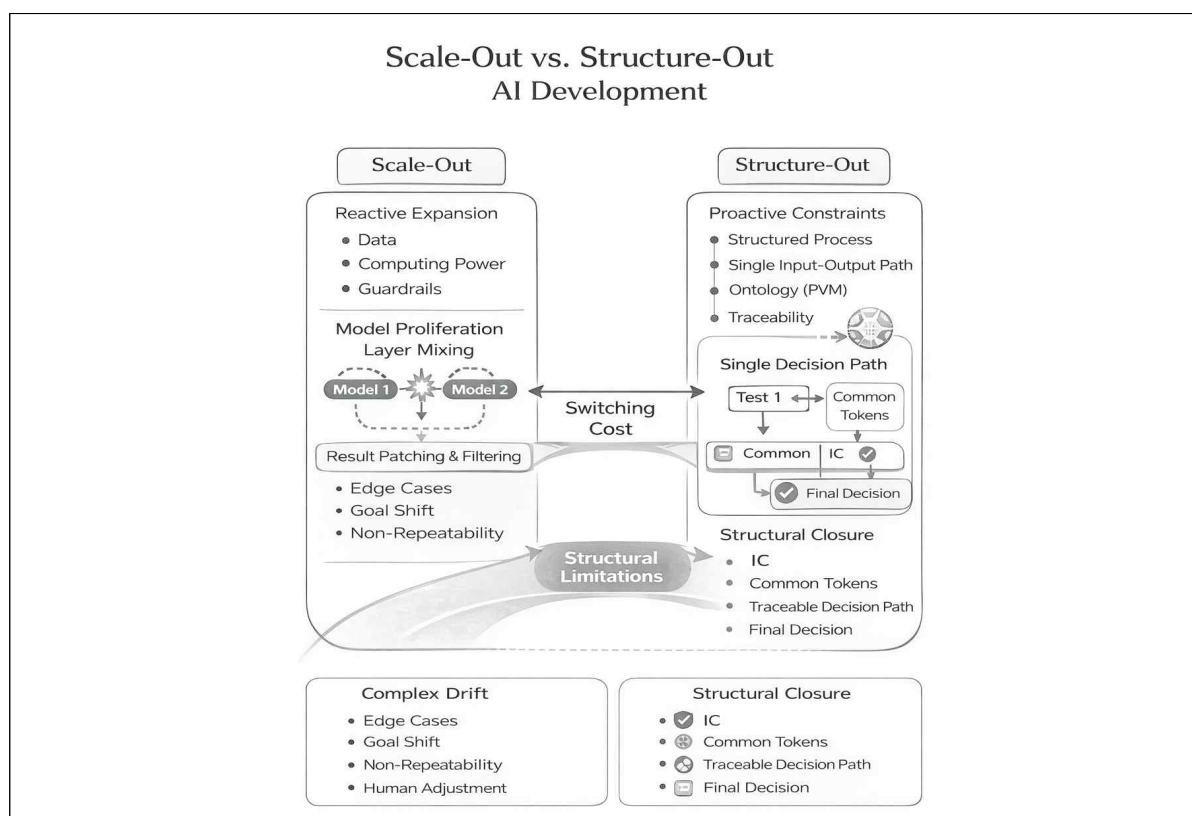
따라서 환각을 제거하는 것이 목표가 되면, 결심이 언제 어떻게 발생하는지에 대한 구조적 질문은 뒷전으로 밀린다. 환각은 줄어들 수 있어도, 결심 경로가 열려 있다면 다른 형태의 "검증되지 않은 연결"이 결론으로 승격될 수 있기 때문이다. 본 연구는 환각을 주요 표적으로 삼는 접근이 근본 원인을 회피한다고 본다.

### 2.3.2 승격 경로가 열려 있을 때의 위험

환각이 치명적인 이유는 승격 경로가 열려 있을 때만 성립한다. 승격 경로란, 중간 산출물이 결심 단계의 절차를 통과하지 않고도 결론으로 반영될 수 있는 통로를 의미한다. 이 통로가 존재하면 시스템은 불확실성의 처리를 "격리와 보류"가 아니라 "결론의 생산"으로 대체한다. 그 결과 환각은 결론 형태로 재구성되어 운영 의사결정에 유입된다.[6],[14]

따라서 해결은 환각을 "없애는 것"이 아니라, 환각이 결론으로 올라가지 못하도록 결심 경로를 봉인하는 것이다. 이는 성능 개선과 독립적인 구조적 처방이며, 본 논문이 제시하는 구조적 단합의 핵심 동기다.

## 2.4 규칙 증식은 해결이 아니라 은폐다



[그림 2-1. 스케일 확장 방식과 구조 확장 방식의 비교]

#### 2.4.1 예외의 예외 구조

실무에서는 실패에 대응해 규칙을 추가하는 방식이 반복된다. 특정 상황에서 실패했다면 그 상황을 예외 처리하고, 다음에는 통과시키기 위한 조건을 덧붙인다. 그러나 이 방식은 필연적으로 예외의 예외를 낳는다. 규칙이 늘어날수록 적용 범위는 분절되고, 경계 영역은 확대된다. 경계가 확대될수록 시스템은 더 자주 “이 경우는 어느 규칙이 적용되는가”라는 메타 문제에 직면한다.[14]

이 시점에서 시스템은 문제를 해결하지 못한다. 단지 실패를 다른 형태로 전치한다. 규칙이 늘어날수록 운영자는 실패를 규칙 위반으로 보지 않고 “새 규칙이 필요한 사례”로 인식하게 된다. 이는 실패를 관측 가능한 자산으로 남기지 못하게 만든다.

#### 2.4.2 경계 붕괴와 충돌

규칙 증식이 위험한 이유는 규칙의 수가 많아서가 아니라, 경계가 붕괴하기 때문이다. 계측 단계, 라벨 단계, 결심 단계가 분리되지 않은 채 규칙이 투입되면, 규칙은 각 단계의 역할을 침범한다. 예를 들어 라벨 단계의 산출물이 결심 기준처럼 취급되거나, 결심 기준이 계측 단계로 역류하는 순간, 시스템은 단계적 의미를 잃고 ‘혼합 규칙 시스템’이 된다.[14]

혼합 규칙 시스템에서는 충돌이 불가피하다. 충돌은 보통 암묵적 우선순위나 운영자의 재량으로 해결된다. 그러나 이러한 해결은 기록되지 않으며, 동일 조건에서 동일 결론을 재현하기 어렵게 만든다. 즉, 규칙 증식은 실패를 줄이기보다 재현성을 붕괴시킨다.

#### 2.4.3 실패의 삭제와 기록 오염

규칙이 실패를 “처리”하는 순간, 실패는 기록에서 삭제된다. 이는 운영 관점에서 가장 치명적인 결과다. 실패가 남지 않으면 원인을 추적할 수 없고, 축적이 없으면 예측 시스템은 개선되지 않는다. 특히 예측 시스템에서 실패는 보정으로 덮어서는 안 되는 관측 자산이다. 규칙 증식은 실패를 통제하는 것이 아니라, 실패를 ‘비가시화’하여 운영을 불능 상태로 만든다.[14]

이 때문에 본 연구는 규칙 증식이 불규칙을 규칙으로 통제하는 방법이라는 통념을 비판한다. 불규칙은 규칙으로 흡수되는 순간 오히려 더 큰 오염으로 돌아온다. 따라서 공통사항은 규칙으로 처리하되, 불규칙은 실패로 남겨 판례로 축적하는 규칙 비증식성 원리가 필요하다.

### 2.5 KG 및 스코프 제한의 한계

#### 2.5.1 스코프 제한의 장점

KG 기반 스코프 제한은 시스템이 말할 수 있는 범위를 명시적으로 한정한다. 이는 임의의 연결 생성을 억제하며, “모르는 것을 모른다고 말할 수 있는 조건”을 제공한다. 책임성과 재현성이 중요한 상황에서 스코프 제한은 유효한 운영 전략이 될 수 있다.

### 2.5.2 결심 오염과의 분리 실패

그러나 스코프 제한이 곧 결심 안정화를 보장하지는 않는다. 스코프 안에서 생성된 산출물이 언제 결론으로 확정되는지, 어떤 절차를 통과해야 하는지에 대한 규정이 없다면, 조기 결정과 사후안정화는 그대로 발생한다. 다시 말해 KG는 “무엇을 말할 수 있는가”를 제한할 수 있지만, “언제 결심할 수 있는가”를 자동으로 통제하지 않는다.

이 한계를 간과하면 KG가 있음에도 불구하고 결심 오염은 발생한다. 결과는 스코프 내부의 정보로만 구성되었기 때문에 겉보기에는 더 신뢰할 수 있어 보이지만, 결심이 허용된 절차를 거치지 않았다면 책임성과 재현성은 확보되지 않는다.

### 2.5.3 결심 경로 봉인의 필요성

따라서 스코프 제한은 결심 경로 봉인을 대체하지 못한다. 본 연구는 KG를 부정하지 않는다. 다만 결심 경로 봉인이 선행되지 않으면 KG는 불충분하다고 본다. 결심이 발생하는 위치와 절차가 구조적으로 고정되어야만 스코프 제한의 효과가 운영 안정성으로 연결된다.

## 2.6 사후 안정화라는 위험한 관행

### 2.6.1 보정이 예측을 연출로 바꾸는 지점

사후 안정화는 실무에서 가장 쉽게 정당화되는 관행이다. 결과가 목표 범위에서 벗어나면 “조금만 조정하면 된다”는 유혹이 발생한다. 그러나 이 순간 예측 시스템은 예측 시스템이기를 멈춘다. 결론이 수정되는 순간, 원래의 결심이 무엇이었는지, 왜 그런 결론이 나왔는지, 어떤 조건에서 확정되었는지가 흐려진다.[14],[5]

사후 안정화는 단기적으로 성능이 개선된 것처럼 보이게 만들 수 있다. 그러나 실제로는 실패의 원인을 삭제하며, 다음 실패를 재현 가능한 형태로 남긴다. 이는 운영 안정성이 아니라 운영 착시다.

### 2.6.2 재현성 붕괴의 발생 조건

예측 시스템의 최소 요건은 재현성이다. 동일한 입력과 동일한 조건에서 동일한 결론이 산출되어야 하며, 결론의 근거가 기록되어야 한다. 사후 안정화는 이 최소 요건을 직접 파괴한다. 동일 입력이라도 보정의 정도가 달라질 수 있고, 보정의 기준이 암묵적으로 변형될 수 있으며, 결과는 운영자의 재량에 의해 달라진다.[5],[14]

따라서 사후 안정화는 기술적 선택이 아니라 구조적 위반으로 간주되어야 한다. 본 연구는 출력 불변과 TRACE를 결심 경로 안정화의 필수 조건으로 제시한다.

## 2.7 결심 개념의 부재

### 2.7.1 결심 미정의가 만드는 혼선

대부분의 LLM 시스템은 추론 과정을 설명할 수는 있지만, 결심을 명시적으로 정의하지 않는다. 결심이란 무엇이며, 어떤 조건에서 허용되는지, 결심이 발생하면 무엇이 종료되는지에 대한 규정이 없다. 그 결과 결심은 사건(event)이 아니라 분위기처럼 발생한다.

결심이 분위기처럼 발생하면, 조기결정이 위반으로 판정되지 않고, 사후 안정화가 위반으로 판정되지 않으며, 중간 산출물의 침범이 위반으로 판정되지 않는다. 즉 결심 미정의는 실패를 구조적으로 허용한다.

### 2.7.2 결심 레이어 단일화 필요성

결심을 통제하려면 결심이 발생하는 레이어가 단일화되어야 한다. 계측과 라벨링은 결심을 위한 입력을 제공할 뿐 결론을 판정할 수 없어야 하며, 결심은 오직 지정된 절차 레일을 통과했을 때만 허용되어야 한다. 이것이 결심 경로 봉인의 전제 조건이다. 본 연구는 이를 레이어 분리와 단일 레일 절차로 구현하는 구조적 단함을 제시한다.

## 2.8 소결: 결심 경로 문제의 재정의

### 2.8.1 결심 허용 조건의 필요성

결심은 언제든지 발생할 수 있는 산출이 아니라, 허용되는 전이여야 한다. 이를 위해 결심 허용의 최소 조건이 명시되어야 하며, 조건이 충족되지 않으면 결심은 발생할 수 없어야 한다. 조건이 명시되지 않는 결심은 운영 위험을 구조적으로 내포한다.

### 2.8.2 절차 레일 강제의 필요성

조건만으로는 충분하지 않다. 결심으로 가는 절차는 단일 레일로 고정되어야 한다. 순서의 누락이나 건너뛰기, 필수 계측의 누락, 불일치가 발생하면 즉시 중단되어야 한다. 절차 강제는 결심 경로 오염을 차단하는 실질적 장치다.

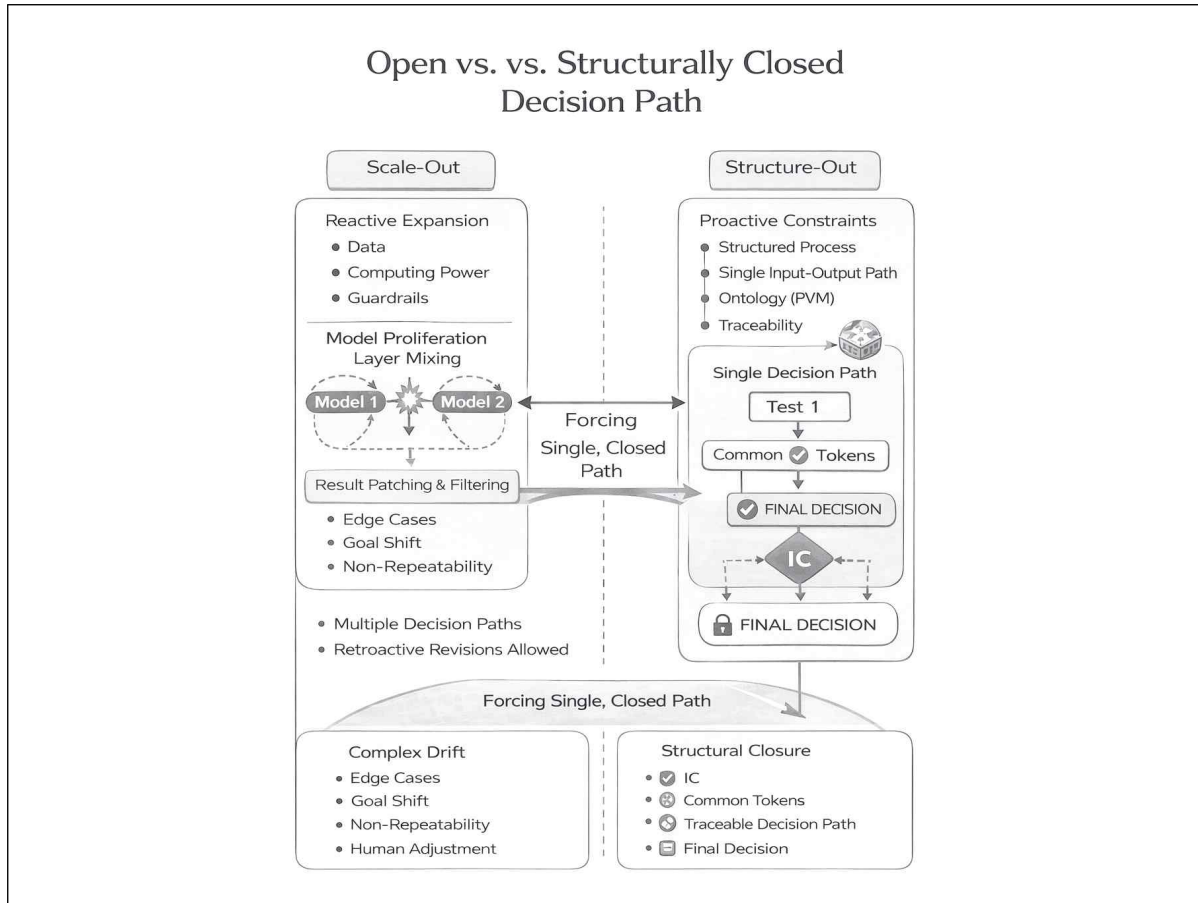
### 2.8.3 구조적 단함으로의 전환

결론적으로, 본 연구가 다루는 LLM 실패는 성능 문제도, 환각 문제도, 규칙 부족 문제도 아니다. 그것은 결심 경로의 구조 문제다. 결심 경로가 열려 있으면 실패는 형태를 바꾸어 반복되고, 운영은 실패를 은폐하는 방향으로 기울어진다. 따라서 필요한 것은 결심이 허용

되는 위치와 절차를 고정하는 구조적 단함이다. 다음 장에서는 이 원리를 구현하는 아키텍처(NCAF·PCM·PVM)를 개요 수준에서 제시한다.

## 제Ⅲ장 구조적 닫힘 아키텍처 개요

### 3.1 설계 목표와 금지 규정



[그림 3-1. 개방형 결정 방식과 구조 확장 방식의 비교]

#### 3.1.1 조기결정 금지

구조적 닫힘의 첫 번째 설계 목표는 조기결정의 구조적 차단이다. 조기결정은 “성급한 답변”이나 “추론 오류”로 환원될 문제가 아니라, 결심이 발생할 수 있는 위치와 절차가 열려 있을 때 필연적으로 발생하는 구조적 현상이다. 예측 시스템에서 조기결정은 치명적인데, 이유는 단순하다. 결론이 확정되는 순간 이후 단계의 계측·라벨·검사 산출물은 결론을 갱신할 권한을 상실하기 때문이다. 즉, 조기결정은 결심 이전 단계가 생성할 수 있는 증거를 구조적으로 무력화한다.

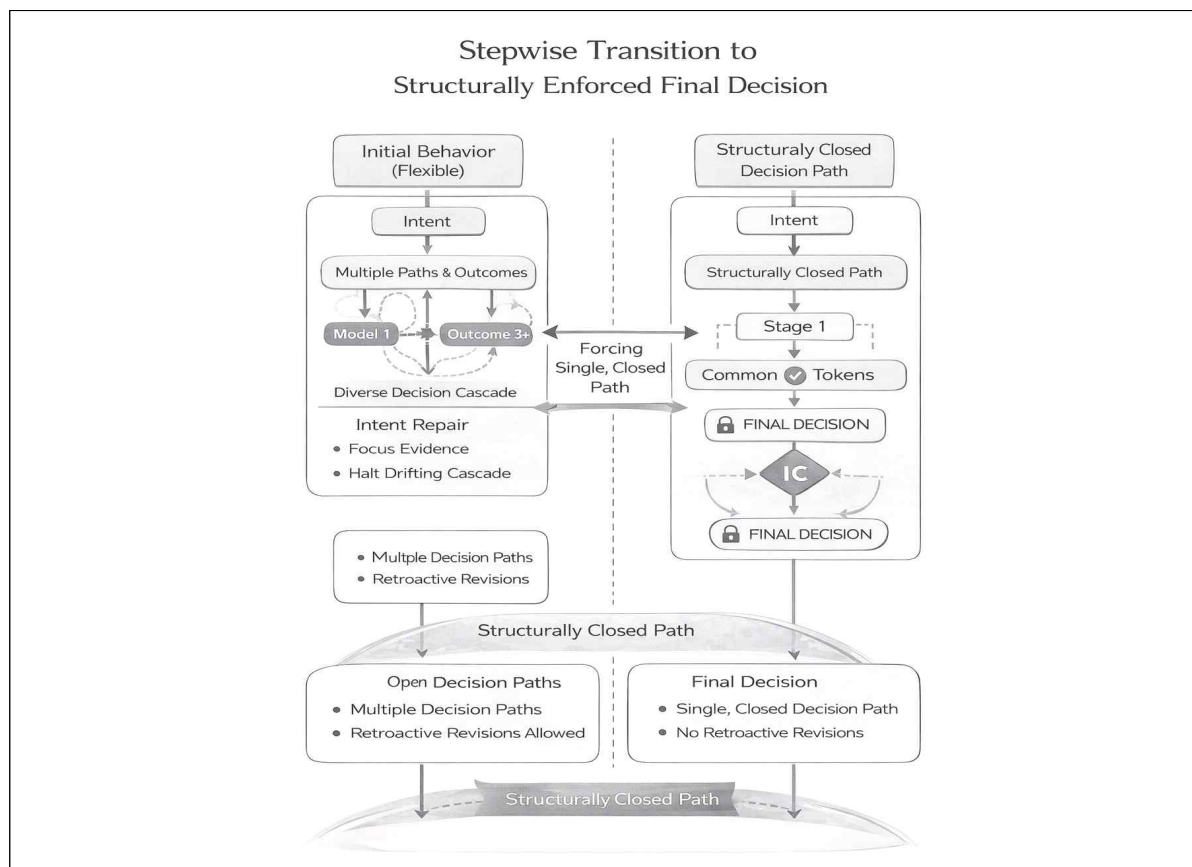
따라서 조기결정 금지는 “주의”나 “가이드라인”으로는 성립하지 않는다. 금지는 실행 규정이어야 한다. 구조적 닫힘은 결심을 하나의 사건(event)으로 정의하고, 그 사건이 허용되기 위한 최소 조건과 통과 절차를 고정한다. 이 조건·절차가 충족되지 않으면 결심은 발생할 수 없으며, 결심이 발생하지 않으면 결론은 출력될 수 없다. 이와 같은 강제는 조기결정을 결과적으로 억제하는 것이 아니라, 조기결정이 발생할 수 있는 경로 자체를 제거한다.

### 3.1.2 사후변경 금지

조기결정이 결심 이전의 증거 생성을 무력화한다면, 사후변경은 결심 이후의 기록을 무력화한다. 사후변경은 결과를 조정하는 행위이기도 하지만, 더 근본적으로는 실패를 관측 불가능하게 만드는 행위다. 예측 시스템은 실패를 제거하는 것이 아니라 실패를 정확히 남김으로써 개선된다. 따라서 구조적 단힘에서 사후변경 금지는 단순 윤리 규정이 아니라 시스템 학습 가능성의 필수 조건이다.

사후변경 금지를 실행 규정으로 만들기 위해서는 출력 불변(immutable output)이 선행되어야 한다. 출력이 불변이면 실패는 실패로 남고, 실패의 조건은 TRACE로 보존된다. 반대로 출력이 가변이면 시스템은 결과를 “맞게 보이도록” 만들기 쉬워지고, 실패는 원인 분석의 대상이 아니라 숨겨야 할 항목이 된다. 구조적 단힘은 이 경로를 원천 차단한다. 결심 이후의 결과는 수정되지 않으며, 수정이 필요한 상황은 결심 이전 단계의 규정 위반 또는 입력 정의의 문제로 귀속된다.

### 3.1.3 결심 레이어 단일화



[그림 3-2. 강제 격리를 포함한 구조적 단힘 결정 경로]

구조적 단힘의 세 번째 목표는 결심 레이어의 단일화이다. 결심 레이어 단일화는 기술 구현의 편의가 아니라, 재현성의 필수 요건이다. 결심이 여러 단계에서 부분적으로 발생하거나,



중간 단계가 사실상 결론을 강제할 수 있는 구조라면, 시스템은 입력이 동일하더라도 내부 경로에 따라 결론이 달라질 수 있다. 이 경우 운영자는 결과를 설명할 수는 있어도, 동일한 조건에서 동일한 결론을 재현하기 어렵다.

단일화의 핵심은 “결심은 오직 결심층에서만 가능하다”는 강제다. 계측층과 라벨층은 결심을 위한 증거를 생성할 뿐, 결심을 대리할 수 없다. 결심층은 계측·라벨의 결과를 받아 결심 허용 여부를 판단하고, 허용된 경우에만 결론을 산출한다. 이때 결심층의 출력 절차는 고정된 순서로만 발생한다. 이 구조는 결심 경로의 분기를 제거하고, 결심이 발생한 위치를 추적 가능하게 만든다.

### 3.2 레이어 분리 원칙(NCAF 개요)

#### 3.2.1 계측 레이어의 역할 고정

구조적 닫힘에서 계측 레이어는 세계를 판단하지 않는다. 계측의 역할은 입력 시퀀스로부터 대칭적이고 누락 없는 측정치를 산출하는 데 한정된다. 계측 레이어가 결론을 유도하려는 순간, 계측치는 근거가 아니라 결론의 도구가 된다. 이는 곧 결심 경로 오염이다.[7],[8]

따라서 계측 레이어는 “정량 산출”만 수행하며, 그 산출은 결심층에 전달될 수 있는 형식으로 정규화된다. 계측 레이어는 판단을 하지 않기 때문에, 불확실성은 불확실성으로 남는다. 이 잔여 불확실성은 결심층에서 격리/검사 절차(IC 등)를 통해 처리된다.

#### 3.2.2 라벨 레이어의 역할 고정

라벨 레이어는 계측치를 해석 가능한 상태 토큰과 패턴 토큰으로 변환한다. 그러나 라벨 레이어 역시 결론을 판정하지 않는다. 라벨은 “세계에 대한 해석”이 아니라 “결심을 위한 상태 표현”이다. 라벨 레이어가 결론 방향으로 의미를 담기 시작하면, 라벨은 상태 토큰이 아니라 결론 토큰으로 변질된다.[7],[8]

이때 가장 위험한 현상은 라벨이 결론을 대리하면서도 책임을 지지 않는 구조가 형성된다는 점이다. 라벨 레이어는 결론의 근거로 호출되지만, 결론을 내린 주체는 아니라고 주장할 수 있다. 구조적 닫힘은 이 모호성을 제거한다. 라벨은 결심층 입력으로만 기능하며, 결론은 결심층에서만 산출된다.

#### 3.2.3 결심 레이어의 유일성

결심 레이어는 유일한 결론 산출 레이어다. 결심 레이어가 유일하다는 말은, 결론으로 이어지는 모든 경로가 이 레이어를 반드시 통과한다는 뜻이다. 따라서 결심 레이어에는 두 가지가 동시에 요구된다. 첫째, 결심 허용 최소 조건을 명시할 것. 둘째, 그 조건이 충족되지 않으면 결심을 발생시키지 않을 것.[7],[8]

여기서 결심 레이어는 “더 똑똑한 추론”을 수행하는 층이 아니다. 결심 레이어의 본질은 ‘허용과 금지’를 집행하는 층이다. 즉, 결심 레이어는 추론의 자유를 억제하는 것이 아니라, 혼선의 경로를 차단함으로써 추론이 자유롭게 산출되더라도 결론의 승격은 규정에 의해 제어되도록 만든다.

### 3.2.4 레이어 침범 금지 규정

레이어 분리를 실질적으로 작동시키기 위해서는 침범 금지 규정이 필요하다. 침범이란 다음을 포함한다. (1) 계측/라벨이 결론을 암묵적으로 결정하는 경우, (2) 결심층이 계측/라벨의 정의를 사후적으로 바꾸는 경우, (3) 운용 중 편의상 절차를 건너뛰는 경우. 침범이 허용되면 구조적 닫힘은 선언에 머무르고, 운영에서는 붕괴한다.[7],[8]

따라서 침범 금지 규정은 “원칙”이 아니라 “중단 조건”으로 구현되어야 한다. 필수 계측이 누락되거나, 절차 순서가 위반되거나, 불일치가 탐지되면 결심은 허용되지 않는다. 이 규정이 바로 공백 봉인의 구현이다. 공백이 존재하는 것은 허용되지만, 공백이 결론을 만들게 하는 것은 금지된다.

## 3.3 결심 허용 규정(PCM 개요)

### 3.3.1 결심 허용의 최소 조건

PCM은 결심이 발생할 수 있는 최소 조건을 정의한다. 여기서 “최소”란 결심을 쉽게 허용한다는 의미가 아니라, 결심을 허용하기 위해 반드시 통과해야 하는 하한선을 의미한다. 최소 조건이 없다면 결심은 항상 가능해지고, 그 결과 시스템은 조기결정과 결론 승격을 반복한다.[1],[2]

최소 조건은 두 종류로 구성된다. 하나는 형식 조건이다. 계측과 라벨이 누락 없이 제공되었는지, 절차 순서가 유지되었는지, 불일치가 없는지 확인한다. 다른 하나는 내용 조건이다. 현재 상태가 결심을 허용할 정도로 충분히 정합적인지, 혹은 격리/검사가 필요한 불확실성을 포함하는지 판정한다. 이때 중요한 점은, 불확실성이 존재한다고 해서 곧바로 결심이 불가능한 것이 아니라, 불확실성을 처리하는 절차(IC 등)가 선행되어야 한다는 것이다.

### 3.3.2 절차 레일(순서 강제)

구조적 닫힘에서 절차는 “권고”가 아니라 “레일”이다. 레일이라는 표현은, 결심이 특정 순서를 벗어나면 더 이상 진행될 수 없음을 의미한다. 절차 레일 강제는 운영 편의에 의해 쉽게 훼손될 수 있으므로, 위반 시 중단되는 형태로 구현되어야 한다.[1],[2]

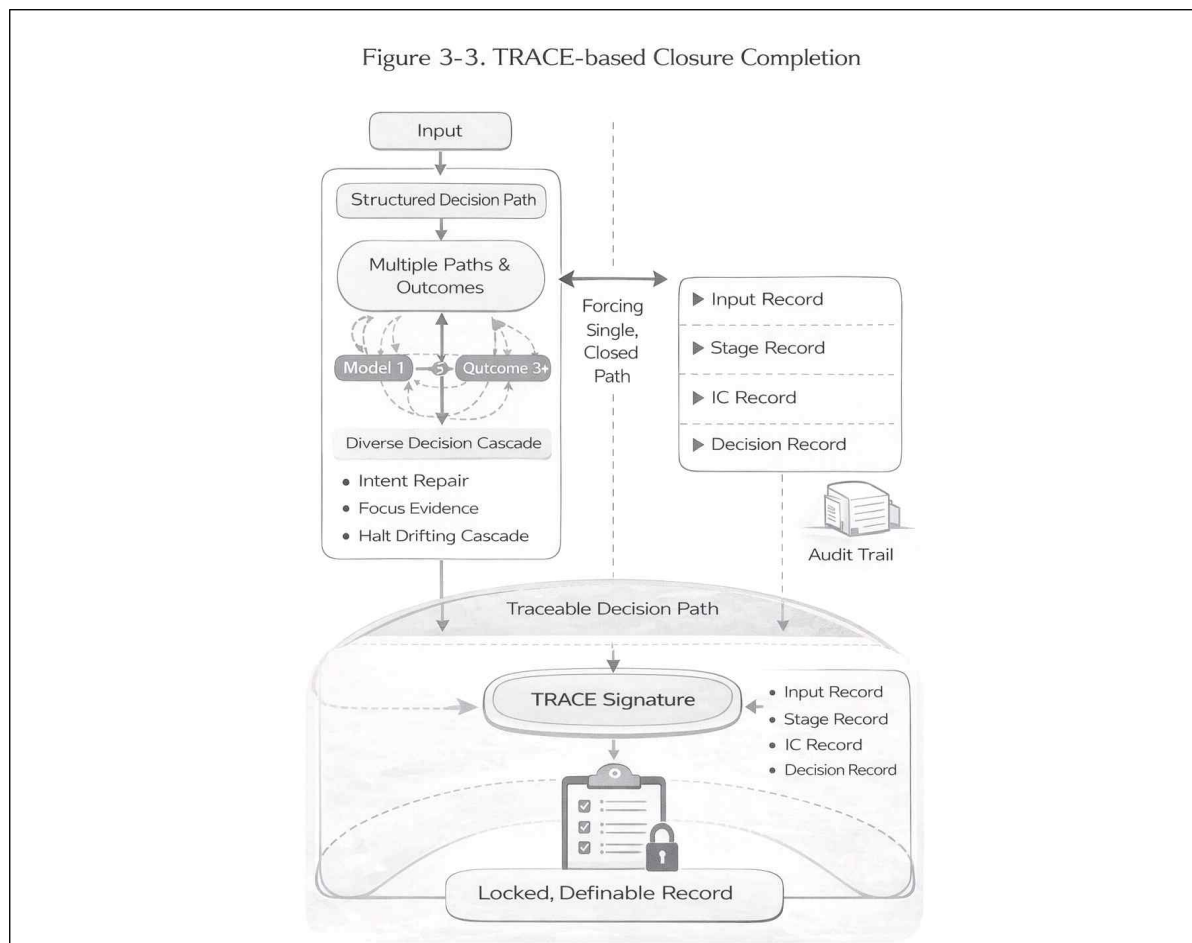
절차 레일의 목적은 두 가지다. 첫째, 결심이 발생한 과정이 추적 가능해진다. 둘째, 조기결정이 구조적으로 불가능해진다. 결심은 레일의 마지막에서만 발생하며, 레일의 중간 단계는 결론을 산출할 권한이 없다. 따라서 절차 레일 강제는 결심 경로 단일화의 실행 형태다.

### 3.3.3 중단 규정(누락/불일치/순서 위반)

중단 규정은 구조적 닫힘의 핵심 안전장치다. 누락, 불일치, 순서 위반이 발생했을 때도 결론을 내리도록 허용하면, 시스템은 “불완전한 근거로 결론을 내리는 구조”를 갖게 된다. 이는 곧 결심 경로 오염이다. 따라서 중단 규정은 예외 처리의 대상이 아니라, 예측 시스템의 최소 윤리이자 최소 공학이다.[1],[2]

중단은 실패를 숨기지 않는다. 실패를 실패로 남긴다. 이 원리는 후속 장에서 제시되는 판례 축적(PVM)의 토대가 된다. 즉, 중단 규정은 운영을 멈추기 위한 장치가 아니라, 실패를 기록 가능한 상태로 보존하기 위한 장치다.

## 3.4 TRACE와 책임성



[그림 3-3. 실행기록(TRACE)에 기반한 구조적 닫힘의 완결]

### 3.4.1 출력 불변 기록

TRACE는 결심 이전과 이후의 실행 흔적을 불변으로 기록하는 구조다. 출력 불변이 없다면 TRACE는 의미를 잃는다. 결과가 사후적으로 변경되면, TRACE는 “왜 그런 결과가 나왔는지”를 설명하는 문서가 아니라, “변경된 결과를 정당화하는 서사”로 전락한다. 구조적 닫힘은 TRACE가 서사가 되지 않도록 출력 불변을 전제한다.[4],[5]

### 3.4.2 재현성 최소 조건

재현성은 동일 입력에서 동일 출력을 보장하는 것만을 의미하지 않는다. 재현성의 최소 조건은 결심이 발생한 절차와 조건이 동일하게 재현되는 것이다. 따라서 구조적 단힘은 출력의 동일성을 넘어, 결심 경로의 동일성을 보장하려 한다. 이는 결심 레이어 단일화와 절차 레일 강제, 중단 규정이 함께 작동할 때 가능해진다.[4],[5]

### 3.4.3 운영 로그 구조화

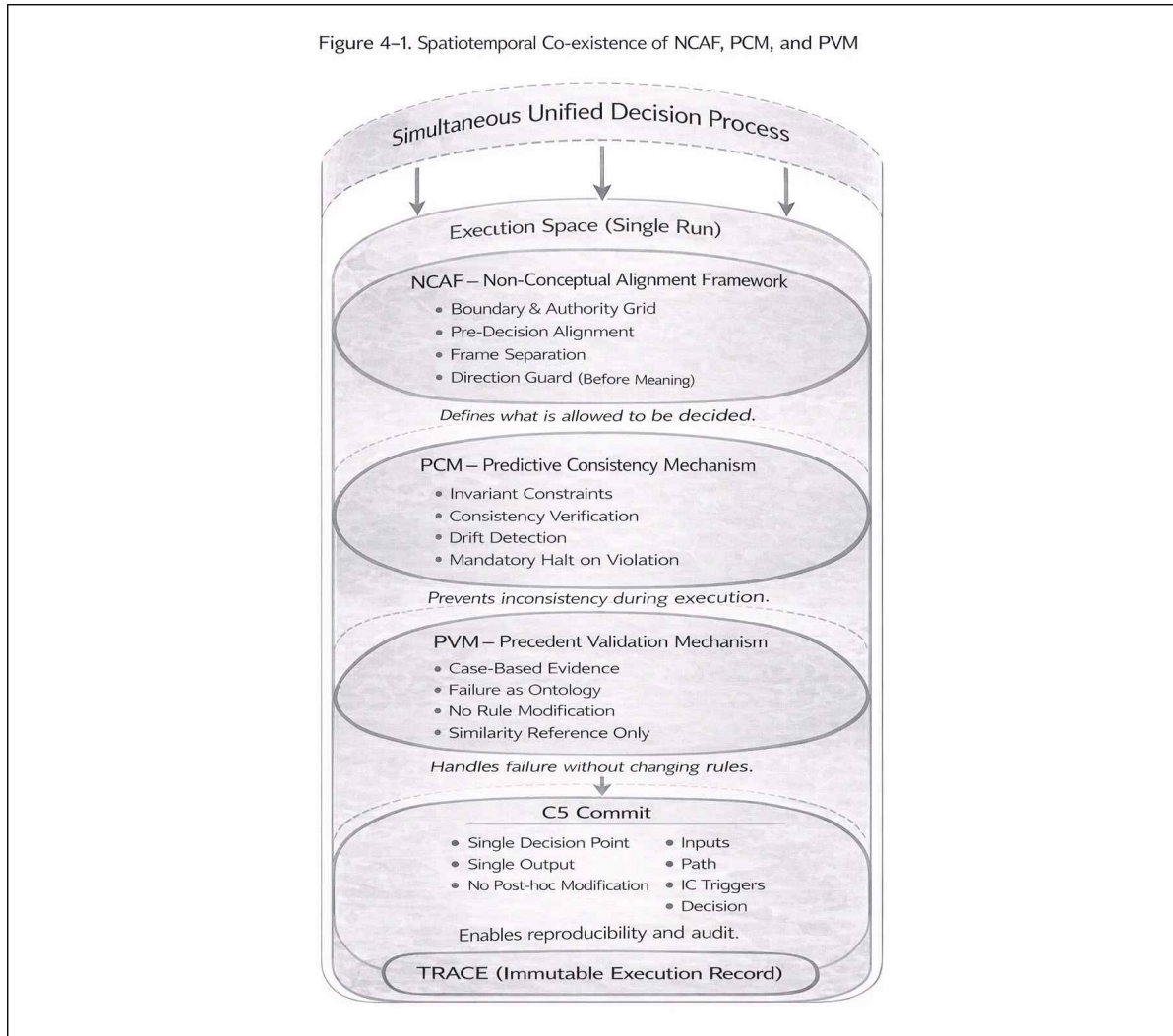
TRACE는 운영 로그의 구조화를 요구한다. 단순한 텍스트 로그나 서술형 기록은 결심 경로를 추적하기 어렵다. 구조적 단힘의 로그는 결심 레일의 단계별 산출물을 고정된 필드로 기록해야 하며, 누락과 위반을 명시적으로 드러내야 한다. 이러한 구조화는 실패를 숨기지 않고, 실패를 판례로 축적하기 위한 입력 형식을 제공한다.[4],[5]

## 3.5 장 소결

본 장은 구조적 단힘 아키텍처의 핵심을 개요 수준에서 제시하였다. 구조적 단힘은 성능 향상이나 규칙 증식이 아니라, 결심이 발생하는 위치와 절차를 고정함으로써 조기결정과 사후변경을 차단한다. 이를 위해 레이어 분리(NCAF), 결심 허용 규정 및 절차 레일 강제(PCM), 출력 불변과 TRACE 기록을 결합한다. 다음 장에서는 NCAF·PCM·PVM을 코어 기술로서 명세하고, 통합 실행 구조에서 경계 조건과 오염 차단 지점을 보다 구체적으로 제시한다.

## 제IV장 코어 기술 명세: NCAF·PCM·PVM의 역할 분리와 통합 실행 구조

### 4.1 장 개요와 명세의 목적



[그림 4-1. NCAF·PCM·PVM의 시공간적 동시 정합 구조]

#### 4.1.1 명세 대상의 한정

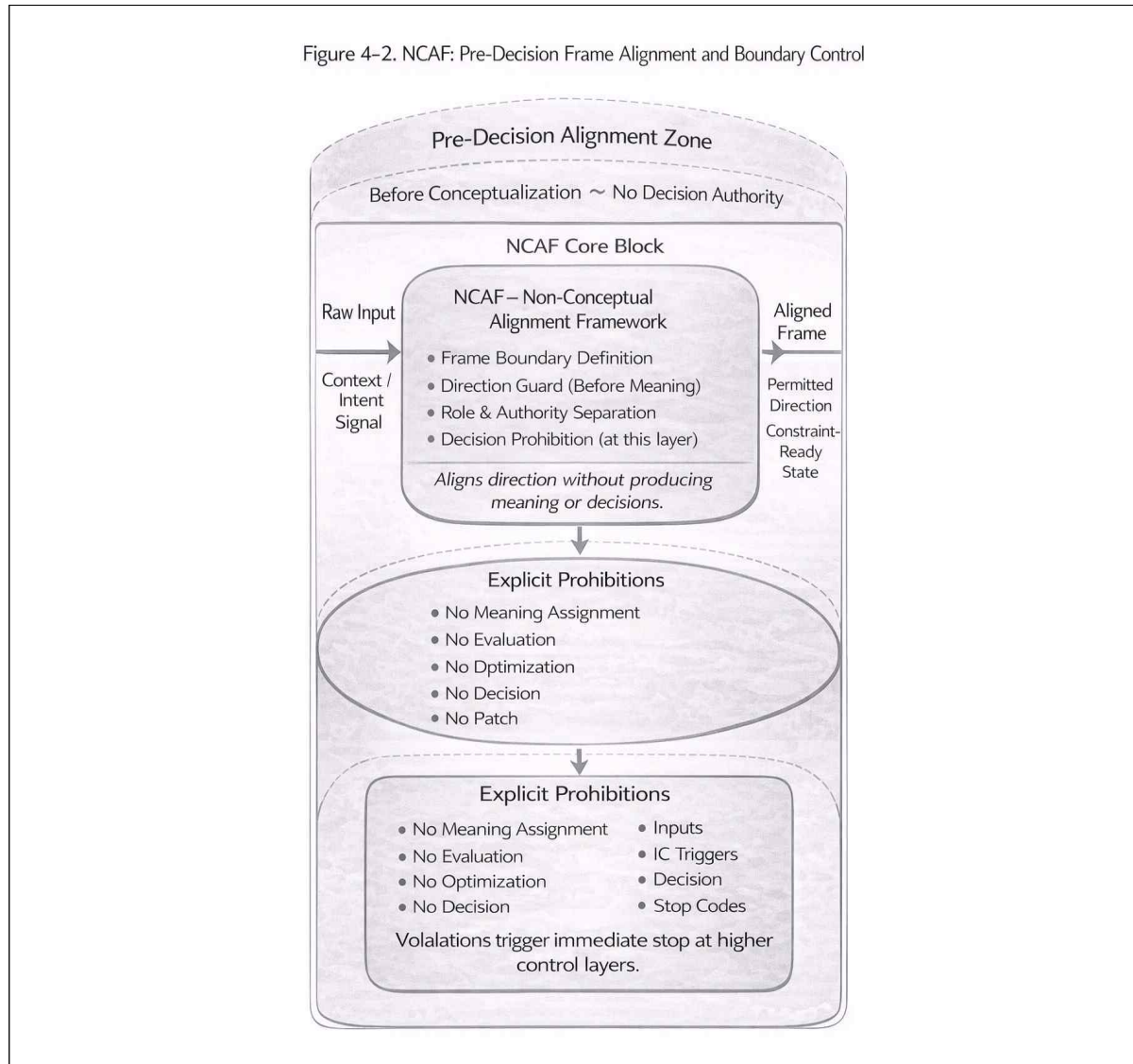
본 장은 구조적 단함을 구현하는 코어 기술로서 NCAF, PCM, PVM을 명세한다. 여기서 명세의 대상은 “좋은 설계의 철학”이 아니라 실행 구조상 각 구성요소가 수행할 수 있는 행위와 수행할 수 없는 행위를 규정하는 것이다. 즉, 본 장은 성능을 높이기 위한 기법 목록이 아니라, 결심 경로 오염을 차단하기 위한 역할 분리 규격이다.

이 명세는 두 가지 이유로 필요하다. 첫째, 구조적 단함이 선언에 머무르지 않기 위해서는 레이어 간 권한 경계가 기계적으로 고정되어야 한다. 둘째, 향후 운용 과정에서 판례가 축적되더라도 규칙이 무한히 증식하지 않기 위해서는 “무엇이 수정 가능하고 무엇이 수정 불가능한가”가 사전에 고정되어야 한다.

#### 4.1.2 명세의 논리적 지위

본 논문에서 명세는 실증 이전에 선행한다. 제5장과 제6장의 실증은 본 장의 명세가 실제 구현과 운용에서 어떻게 유지되는지를 보여주는 단계다. 따라서 제4장은 실증을 위한 해석 프레임이 아니라, 실증이 따라야 하는 규율의 선언이며, 제5·6장은 이 선언을 위반하지 않는 형태로만 서술되어야 한다.

## 4.2 NCAF: 인지 정렬 프레임과 레이어 권한 체계



[그림 4-2. 사전 결심 프레임 정렬 구조(NCAF)]

#### 4.2.1 NCAF의 정의

NCAF는 예측 시스템을 단일 추론 블록으로 보지 않고, 예측·상태·패턴·결심의 계층적 흐름으로 구성된 인지 정렬 구조로 본다. 여기서 “정렬”이란 외부 목표에 맞춘 출력 조정이 아니라, 내부적으로 결심이 발생할 수 있는 경로를 단일화하고 단계별 산출물의 의미를 안정화하는 것을 의미한다.[7],[8]

NCAF의 핵심 기여는 레이어의 분리다. 레이어 분리는 구성요소를 나누는 설계 미학이 아니라, 결심 책임을 한 지점에 귀속시키기 위한 필수 조건이다. 계측과 라벨이 결론을 침범할 수 없도록 권한을 박탈하는 동시에, 결심층에는 허용·차단의 집행 책임을 부여한다.

#### 4.2.2 레이어별 권한과 금지 행위

NCAF에서 각 레이어의 권한은 다음과 같이 고정된다.

첫째, 계측 레이어는 수치·관측량을 산출할 수 있으나 세계 판정을 할 수 없다. 즉, 방향·진동폭 결론을 직접 또는 간접으로 강제하는 산출을 생성할 권한이 없다. 계측은 오직 “측정”이며, 측정은 결론이 아니다.[7],[8]

둘째, 라벨 레이어는 계측량을 상태 토큰과 패턴 토큰으로 변환할 수 있으나 결론을 판정할 수 없다. 라벨은 해석 가능한 표현일 뿐 결심이 아니다. 라벨이 결론을 암시하는 순간, 결심은 라벨 단계로 역류한다.

셋째, 결심 레이어는 유일하게 결론을 산출할 수 있으나, 계측 정의나 라벨 정의를 사후적으로 변경할 수 없다. 결심층이 중간층의 정의를 바꾸기 시작하면, 결심은 일관되게 재현될 수 없으며 로그는 서사로 변질된다.

이와 같은 권한·금지 규정이 고정되어야만, 구조적 단힘은 “결론을 내지 말라”는 태도 규정이 아니라 “결론을 낼 수 없는 구조”로 구현된다.

#### 4.2.3 레이어 분리의 귀결: 책임성과 재현성

레이어 분리의 귀결은 명확하다. 결심은 오직 결심층에서만 발생하므로 책임은 결심층에 귀속된다. 또한 계측·라벨은 결심의 입력으로만 기능하므로, 동일 입력에서 동일 계측·라벨이 생성되면 결심 경로의 재현성이 확보된다. 즉, NCAF는 “정답률”이 아니라 “결심 경로의 재현성”을 실무 시스템의 핵심 성질로 제안한다.[7],[8]

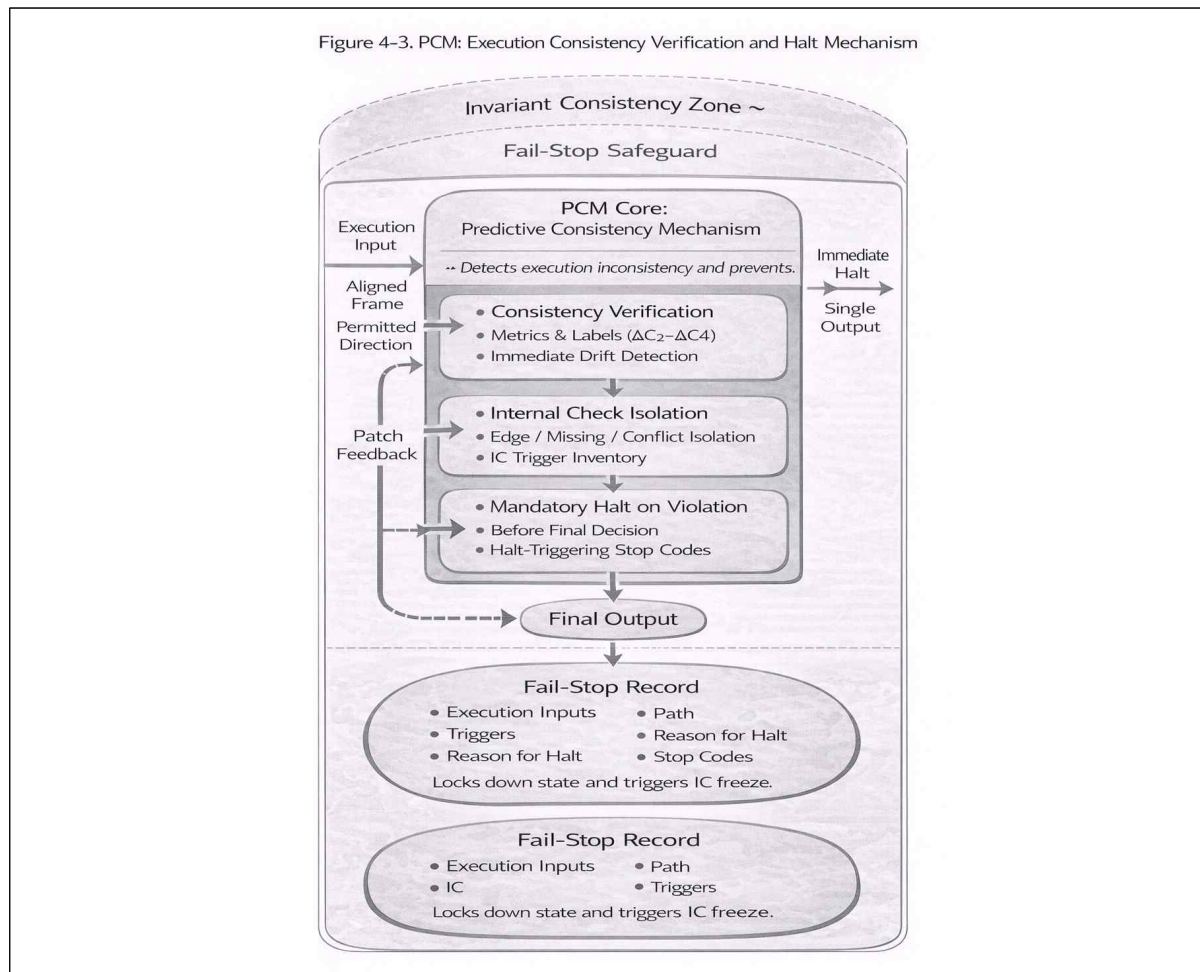
### 4.3 PCM: 결심 허용 제약(Principled Constraint Model)

#### 4.3.1 PCM의 정의: 결심은 허용되는 전이다

PCM은 결심을 “언제든 발생할 수 있는 출력 이벤트”가 아니라, 특정 조건을 통과했을 때만 발생 가능한 전이(transition)로 정의한다. 이는 구조적 단힘의 핵심이며, 결심 경로 오염을 차단하는 직접적 장치다.[4],[5]

중요한 점은 PCM이 결론을 더 잘 맞히기 위한 추론 규칙이 아니라는 것이다. PCM은 결론을 내릴 자격이 있는지 없는지를 판단하는 규율이다. 다시 말해 PCM은 성능 장치가

아니라 안전 장치이며, 예측 시스템을 예측 시스템으로 남게 하는 최소 규범이다.



[그림 4-3. 일관성 검증 및 중단 메커니즘(PCM)]

#### 4.3.2 결심 허용 최소 조건(Minimum Admissibility)

PCM은 결심 허용의 최소 조건을 두 축으로 고정한다.

첫째, 형식 조건이다. 필수 계측의 누락이 없어야 하며, 라벨 산출물의 일관성이 유지되어야 하고, 절차 순서 위반이 없어야 한다. 형식 조건이 충족되지 않으면, 결심은 “근거 부족 상태에서의 결론 승격”이 된다.[4],[5]

둘째, 내용 조건이다. 현재 상태가 정합성 기반의 결심을 허용하는지, 혹은 혼선 구간으로서 격리/검사가 필요한지 판정한다. 내용 조건은 “확신이 높으니 바로 결론”이 아니라, “혼선 가능성이 있으니 결심을 지연하고 검사를 수행”하는 방향으로 작동해야 한다. 이때 PCM은 결심의 속도를 높이는 것이 아니라, 결심의 정당성을 보호한다.

#### 4.3.3 절차 레일 강제(Sequence Lock)

PCM의 핵심 집행 방식은 절차 레일 강제다. 결심은 반드시 규정된 순서를 따라야 하며, 중간 단계를 생략할 수 없다. 절차가 레일로 고정되지 않으면 운용자는 편의상 단계를 건너



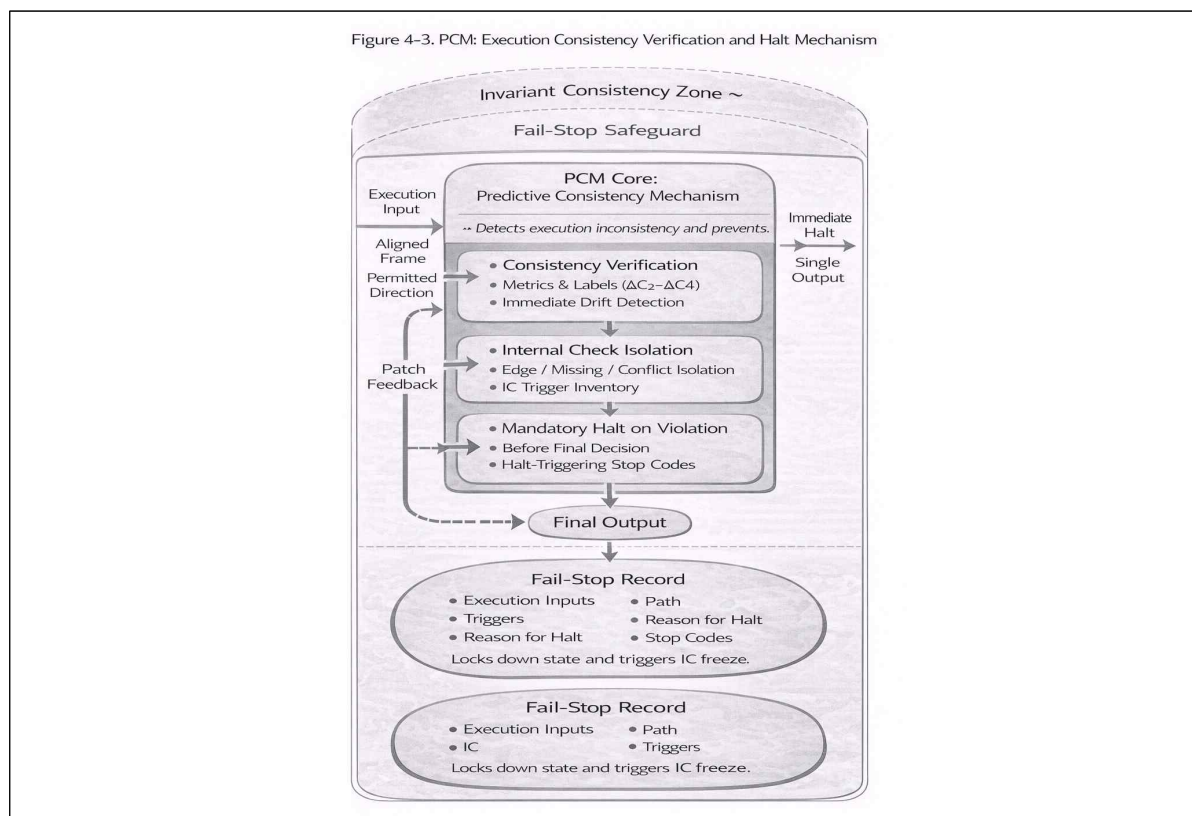
뛰게 되고, 그 순간 결심 경로 오염은 구조적으로 재도입된다.[4],[5]

PCM이 강제하는 레일은 "결심을 빠르게 내리기 위한 절차"가 아니라, 결심이 허용될 수 있는 조건과 권한 경계를 실행 중에도 유지하기 위한 최소 규범이다. 특히 누락·불일치·순서 위반은 결심을 지연시키는 사유가 아니라, 결심 자체를 금지해야 하는 무결성 위반으로 취급된다.

[Rule: Fail-Stop] 필수 입력/계측/라벨이 누락되거나, 산출물 간 불일치가 탐지되거나, 절차 순서 위반이 발생하면 PCM은 결론을 산출하지 않고 즉시 중단한다. 이 중단은 실패를 "없애는 것"이 아니라 실패를 중단 코드와 TRACE로 기록 가능한 형태로 남기는 것이며, 이후 판례(PVM)의 입력이 오염되지 않도록 하는 필수 장치다.

[Rule: Output Immutability] 결심(FINAL)이 커밋된 이후에는 어떠한 모듈도 결론을 수정/덮어쓰기 할 수 없다. 커밋 이후의 개입은 경고/위험도/범위 조정과 같은 메타 수준의 신호에 한정되며, 결론 변경은 허용되지 않는다. 동일 사안에 대한 재실행은 "수정"이 아니라 새 TRACE로 기록되는 별도 실행으로만 존재한다.

#### 4.4 PVM<sup>7)</sup>: 판례 참조(Past-Case Validation Mechanism)



[그림 4-4. 규칙 수정 없는 판례 기반 검증 구조(PVM)]

7) PVM(판례 참조 메커니즘)은 결심을 '바꾸기' 위한 장치가 아니라, 결심이 허용된 이후 동일·유사 구조의 과거 사례를 제한적으로 참조하여 위험도/경고/예측 범위 조정 같은 메타 신호를 제공하는 장치이다. 본 논문은 판례가 결심 레일을 침범하여 결론을 수정하는 형태의 운용을 금지한다.

#### 4.4.1 PVM의 정의: 규칙이 아니라 판례

PVM은 실패를 규칙으로 흡수하지 않고 판례로 축적하는 메커니즘이다. PVM의 핵심은 규칙 비증식성이다. 판례가 축적되더라도 PCM의 제약 조건이나 NCAF의 레이어 권한은 변경되지 않는다. 판례는 구조를 바꾸지 않고, 구조 위에서만 작동한다.[14]

이 원리는 단순한 운영 선언이 아니라, 규칙 증식이 장기적으로 시스템을 붕괴시킨다는 경험적 사실에 근거한다. 규칙이 늘어날수록 경계가 붕괴하고 충돌이 증가하며, 무엇보다 실패가 기록에서 삭제된다. PVM은 이 경로를 차단한다.

#### 4.4.2 작동 시점: PCM 이후

PVM은 PCM에 의해 결심이 허용된 이후에만 작동한다. 즉, 결심 경로가 구조적으로 닫힌 상태에서는 판례가 결론 생성에 개입하지 않는다. 이는 판례가 결심 경로의 봉인을 침해하지 않도록 하기 위한 설계다.[14]

이 시점 제한이 없다면 PVM은 사실상 사후 보정 도구로 오해될 수 있다. 그러나 본 논문에서 PVM은 결론을 수정하는 장치가 아니다. PVM 출력은 경고/위험도/예측 범위 조정 에 한정되며, FINAL 커밋은 PCM에 의해 불변이다. PVM은 현재 상태와 판례 집합의 유사성을 비교하여, 현재 결심이 구조적으로 안정적인지 점검하고, 불확실성이 높은 경우 결심의 위험도를 표기하거나 운용상 경고를 제공하는 역할을 한다. 결론을 바꾸지 않기 때문에 출력 불변 규정과 충돌하지 않는다.

#### 4.4.3 판례 축적과 실패의 자산화

PVM이 작동하려면 실패가 실패로 남아야 한다. 즉, 사후 안정화로 실패를 덮는 순간 판례는 축적될 수 없고, 시스템은 개선될 수 없다. PVM은 실패를 제거하지 않고, 실패를 구조적 좌표로 저장한다. 동일한 구조적 맥락에서 과거에 관측된 결과를 참조함으로써, 시스템은 규칙 증식 없이도 점진적으로 운영 안정성을 높인다.[14]

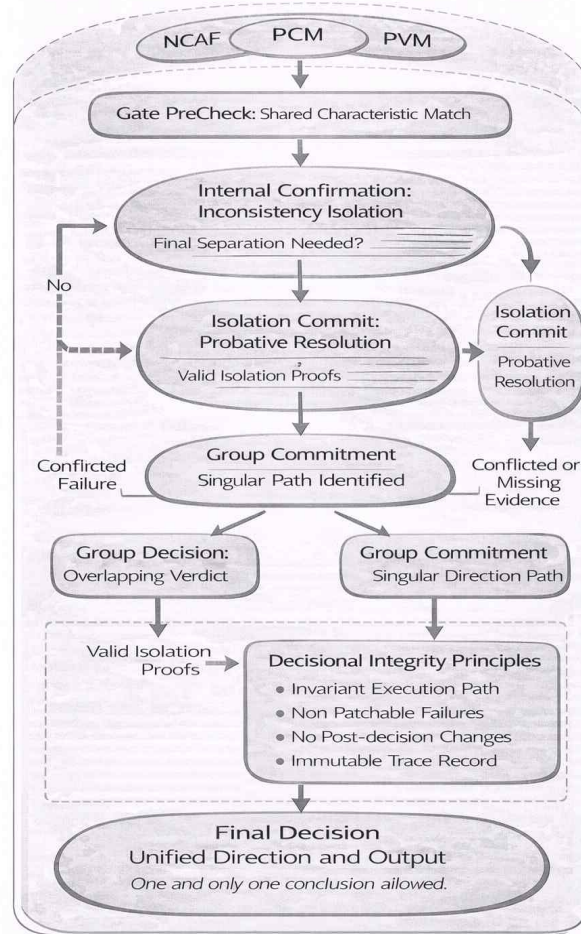
이때 판례의 역할은 "결론 수정"이 아니라 "결심 안정성 검증"이다. 이 구분이 유지되기 때문에, PVM은 사후 보정과 다르다.

### 4.5 통합 실행 구조: 결심 경로 봉인의 완결

#### 4.5.1 통합 파이프라인

NCAF·PCM·PVM은 독립 모듈이 아니라, 결심 경로 봉인을 목표로 결합된 실행 구조다. NCAF는 레이어 권한을 고정하고, PCM은 결심 허용의 절차 레일을 강제하며, PVM은 규칙 비증식성 하에서 실패를 판례로 축적한다. 이 통합 구조에서 가장 중요한 점은 결심이 단일 레이어에서만 발생하며, 그 결심은 허용 조건과 절차를 통과했을 때만 발생한다는 것이다.

Figure 4-5. Single Commitment Flow with Structural Closure

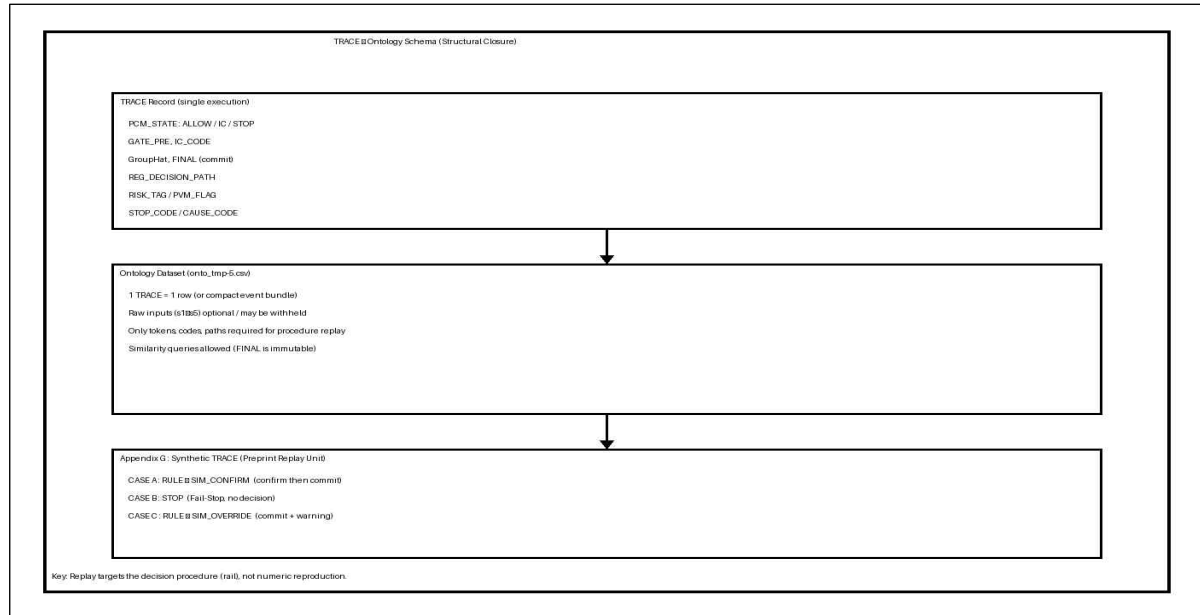


[그림 4-5. 단일 서명과 커밋으로 수렴되는 통합 제어 출력]

Authority Matrix (Corrected)			
Layer	May Decide	May Modify	May Override
NCAF	Constraints		
PCM	Decision Path		
DS Commit	FINAL		
PVM		Risk/Warning	
TRACE		Log Only	

[그림 4-6. 권한 매트릭스(Authority Matrix): NCAF·PCM·PVM·C5 TRACE의 허용/금지 경계 ]

그림 4-6은 1회 실행의 TRACE가 onto\_tmp-5.csv 온톨로지 좌표로 저장되고, 부록 G의 합성 TRACE(3케이스)로 절차 재현 단위를 구성하는 흐름을 나타낸다. 재현의 목표는 수치가 아니라 결심 레일(절차)의 재현이다.



[그림 4-7. TRACE-온톨로지 스키마(TRACE/Ontology Schema): 절차 재현 단위의 기록 구조]

그림 4-7은 1회 실행의 TRACE가 온톨로지 좌표(onto\_tmp-5.csv)로 저장되고, 부록 G의 합성 TRACE를 통해 절차 재현 단위를 구성하는 구조를 나타낸다. 재현의 대상은 수치가 아니라 결심 경로(절차 레일)이다.

#### 4.5.2 오염 차단 지점

오염은 대개 세 지점에서 발생한다. (1) 계측·라벨이 결론을 암묵적으로 강제할 때, (2) 결심 이후 결과가 수정될 때, (3) 실패가 규칙에 흡수되어 기록에서 삭제될 때다. 통합 구조는 각각을 다음과 같이 차단한다. (1)은 레이어 권한 분리, (2)는 출력 불변으로, (3)은 규칙 비증식성과 판례 축적으로 차단된다.

#### 4.5.3 실증과의 연결

제5장은 이 통합 구조가 방향 판단 프로세서에서 설계·연산 차원으로 이미 구현되어 있음을 보여준다. 제6장은 동일한 구조가 진동폭 처리 프로세서의 운용 로그에서 유지됨을 보여준다. 즉, 제4장은 실증의 전제이며, 제5·6장은 제4장의 명세가 위반되지 않았음을 보여주는 단계다.

### 4.6 장 소결

본 장은 구조적 닫힘을 구현하는 코어 기술 NCAF·PCM·PVM을 명세하였다. NCAF는 레이어

권한을 분리하여 계측·라벨이 결론을 침범하지 못하게 하며, PCM은 결심을 허용되는 전이로 정의하고 절차 레일을 강제한다. PVM은 PCM 이후에만 작동하며, 실패를 규칙으로 흡수하지 않고 판례로 축적하여 규칙 비증식성 하에서 운영 안정성을 높인다. 이 통합 실행 구조는 결심 경로 오염을 차단하고, 예측 시스템을 재현 가능하고 책임 가능한 형태로 고정한다.

## 제5장 실증 1: 방향 판단 프로세서의 설계·연산 차원 구조적 닫힘

### 5.1 문제 설정과 I/O 규격

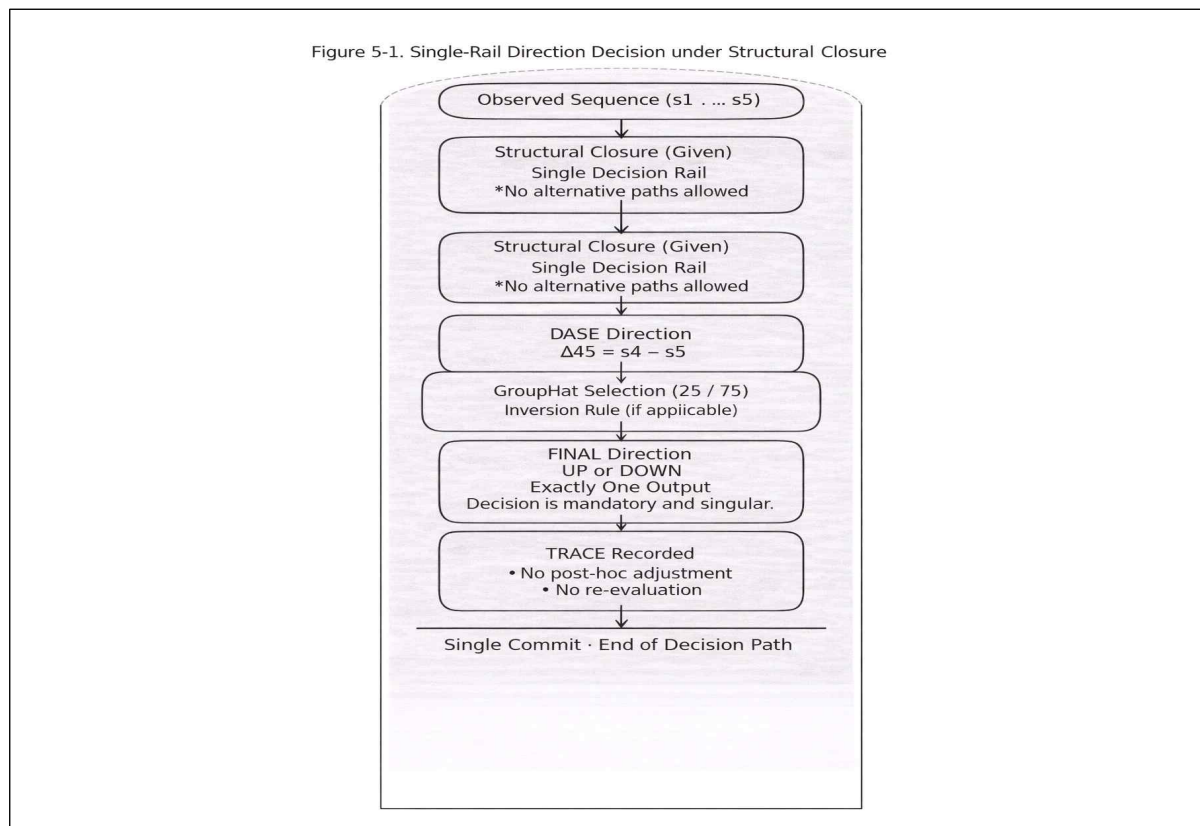
#### 5.1.1 입력 시퀀스와 $\Delta t$ 규정

방향 판단 프로세서는 연속된 상태 시퀀스를 입력으로 받아, 단일 방향 결론을 산출하는 예측 모듈이다. 본 연구에서 방향 판단은 “다음 상태의 방향(UP/DOWN)”을 결정하는 문제로 정의되며, 입력은 시간 간격이 고정된 상태값 시퀀스로 구성된다. 모든 연산은  $\Delta t=1$ 의 동일한 시간 간격을 전제로 수행되며, 시간 간격의 가변성은 허용되지 않는다.

이 규정은 단순한 입력 조건이 아니라, 결심 경로 안정화를 위한 전제다.  $\Delta t$ 가 가변적일 경우, 계측 단계에서 산출되는 값의 의미가 흔들리고, 라벨 단계에서의 상태 해석이 결심 판단에 암묵적으로 개입할 여지가 생긴다. 따라서 입력 시퀀스의 시간 간격 고정은 방향 판단 프로세서의 구조적 닫힘을 위한 최소 요건으로 설정된다.

#### 5.1.2 출력 정의(UP/DOWN)

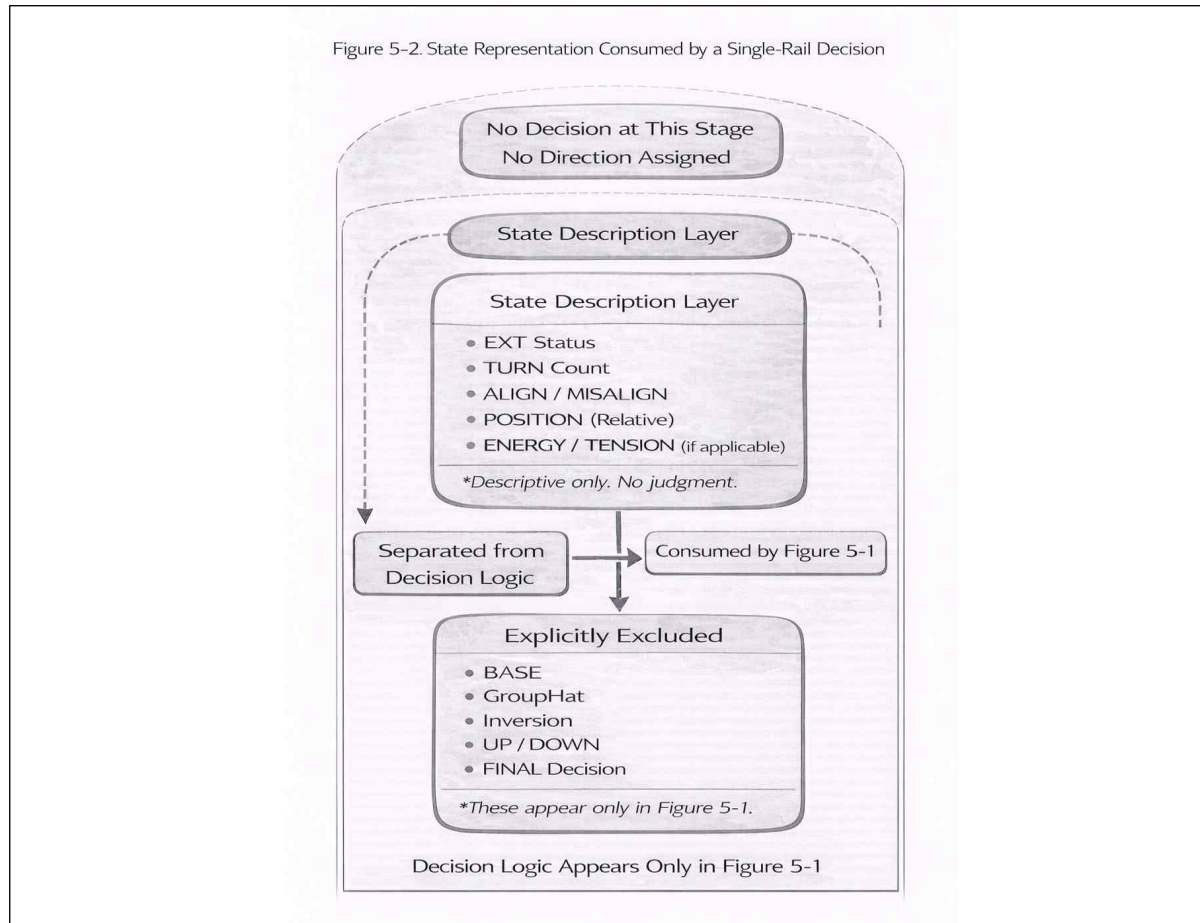
방향 판단 프로세서의 출력은 이진적이다. 최종 출력은 UP 또는 DOWN 중 하나로만 산출되며, 보류, 중립, 미정과 같은 제3의 출력은 허용되지 않는다. 이 규정은 방향 판단을 “확률적 의견 제시”가 아니라 “결심 사건”으로 다루기 위한 것이다.



[그림 5-1.구조적 닫힘 하의 단일 레일 방향 결정]

출력의 단순화는 정보 손실을 의미하지 않는다. 오히려 결심이 발생하는 지점을 명확히 드러낸다. 결심 이전 단계에서는 충분히 다양한 계측과 라벨이 생성될 수 있으나, 결심층에서는 단일 방향만이 허용된다. 이 이분법적 출력 규정은 결심 레이어 단일화의 직접적인 구현이다.

## 5.2 레이어 분리: 계측·라벨과 결심의 분리



[그림 5-2. 단일 레일 결정을 위한 상태 기술 구조]

### 5.2.1 기준1~4의 결정 금지

방향 판단 프로세서는 기준1부터 기준4까지를 계측 및 라벨 전용 레이어로 정의한다. 이 레이어들의 역할은 상태를 측정하고, 패턴과 에너지, 위치와 정합성을 표현하는 토큰을 생성하는 데 한정된다. 이 단계에서는 방향을 판정하거나 결론을 암시하는 어떤 연산도 허용되지 않는다.

이 결정 금지 규정은 선언이 아니라 실행 규칙이다. 기준1~4에서 생성된 산출물은 결심층(기준5)에 전달될 수는 있으나, 그 자체로 방향을 결정할 권한을 갖지 않는다. 만약 기준1~4의 산출물이 방향을 암묵적으로 강제한다면, 결심은 이미 발생한 것이며, 기준5는 형식적인 통과 절차로 전락한다. 구조적 단힘은 이 경로를 명시적으로 차단한다.

### 5.2.2 기준5의 유일 결심권

방향 판단에서 결심은 오직 기준5에서만 발생한다. 기준5는 계측과 라벨 결과를 종합하여 결심 허용 여부를 판단하고, 허용된 경우에만 최종 방향을 산출한다. 이 구조는 결심이 “분산된 판단의 합”이 아니라, 단일 레이어에서 발생하는 사건임을 보장한다.

기준5의 역할은 더 많은 정보를 해석하는 것이 아니라, 결심을 허용할 것인지 차단할 것 인지를 집행하는 것이다. 이때 기준5는 결론을 내리는 주체로서 모든 책임을 갖는다. 계측 이나 라벨 단계는 결론의 근거를 제공할 수는 있지만, 결론의 책임을 대신 질 수는 없다.

### 5.2.3 누락·불일치 중단 규정

기준5로 입력되는 정보에 누락이나 불일치가 존재할 경우, 결심은 즉시 중단된다. 이는 방향 판단 프로세서에서 가장 중요한 안전장치다. 누락된 계측이나 상충하는 라벨이 존재 하는 상태에서 결론을 내리는 것은, 불완전한 근거를 결론으로 승격시키는 행위이기 때문이다.

중단은 실패가 아니다. 중단은 실패를 기록 가능한 상태로 남기는 선택이다. 이 규정이 존재하기 때문에, 방향 판단은 “어쨌든 하나의 방향을 내는 시스템”이 아니라, “결심이 허용되는 조건에서만 방향을 내는 시스템”이 된다.

## 5.3 결심 경로 단일화

### 5.3.1 BASE와 GroupHat의 분리

방향 판단 프로세서에서 BASE는 기준1에서 산출되는 기초 방향 정보다. BASE는 계측 결과 에 의해 결정되며, 그 자체로는 최종 결론이 아니다. GroupHat은 기준5에서 산출되는 세계 판정 토큰으로, BASE를 그대로 사용할지, 반전할지를 결정한다.

이 분리는 매우 중요하다. BASE는 “관측된 경향”이고, GroupHat은 “결심에 허용된 세계 해석”이다. 두 개념을 분리함으로써, 관측과 결심이 동일시되는 것을 방지한다. BASE가 맞았는지 틀렸는지는 사후 문제가 아니라, GroupHat이 어떤 조건에서 어떤 선택을 했는 지가 결심의 핵심이 된다.

### 5.3.2 출력 절차 레일 고정

방향 판단의 출력 절차는 고정된 순서를 따른다. GATE\_PRE → IC → GroupHat → FINAL 의 순서는 임의로 변경될 수 없으며, 단계 생략이나 건너뛰기는 허용되지 않는다. 이 절차 레일은 결심 경로 단일화의 실행 형태다.



절차가 고정되면, 결심이 언제 발생했는지 명확해진다. 또한 결심 이전 단계가 결론을 암시하거나 강제하는 경로가 차단된다. 방향 판단은 이 레일의 마지막에서만 결론을 산출한다.

## 5.4 결심 허용 최소 조건(PCM 적용)

### 5.4.1 REGIME-LOCK

REGIME-LOCK은 현재 상태가 극단 영역인지, 중간 영역인지에 따라 결심 경로를 분기시키는 사전 격리 규정이다. 이 규정은 결심을 빠르게 하기 위한 장치가 아니라, 혼선이 발생하기 쉬운 구간에서 결심을 지연시키기 위한 장치다.[1],[2]

REGIME-LOCK에 의해 특정 상태는 즉시 결심 후보에서 제외되거나, 내부 확인(IC) 절차로 강제 이동된다. 이로써 극단 신호가 결론을 자동으로 유도하는 경로를 차단한다.[11]

### 5.4.2 GATE\_PRE

GATE\_PRE는 결심의 1차 관문이다. 이 단계에서는 결심이 즉시 가능(25/75)한지, 아니면 내부 확인이 필요한지(IC)를 판정한다. 중요한 점은, GATE\_PRE 자체가 결심을 발생시키지 않는다는 것이다. GATE\_PRE는 결심 허용 여부를 가르는 필터일 뿐이다.[1],[2][12]

### 5.4.3 IC

IC는 결심 지연 및 격리 구간이다. IC 구간에서는 추가 증거가 확인되기 전까지 결심이 발생하지 않는다. IC는 결심을 늦추는 장치이지, 결심을 회피하는 장치가 아니다. 이 단계는 혼선 구간에서 조기결정을 구조적으로 차단한다.[1],[2]

### 5.4.4 GroupHat

GroupHat은 IC 또는 GATE\_PRE를 통과한 이후에만 산출된다. GroupHat은 BASE를 유지할지 반전할지를 결정하며, 이 결정이 곧 세계 판정이다. 이 시점에서 결심은 허용되며, 이전 단계로 되돌아갈 수 없다.[1],[2]

### 5.4.5 FINAL

FINAL은 GroupHat과 BASE의 결합 결과다. FINAL이 산출되는 순간 방향 판단은 종료된다. 이 결과는 출력 불변 규정에 따라 수정되지 않으며, 이후의 모든 분석은 이 결론을 기준으로 수행된다.[1],[2]

### 5.4.6 요약

본 절은 방향 판단에서 결심이 발생하는 유일 레이어와, 결심이 허용되기 위한 최소 조건을 정의하였다. 방향 결심은 언제든지 발생할 수 있는 이벤트가 아니라, 엄격하게 정의된

조건과 절차를 통과했을 때만 허용되는 구조적 전이다. 이 설계를 통해 방향 판단은 조기 결정과 오판을 구조적으로 차단하며, 예측 시스템으로서의 안정성을 확보한다.

## 5.5 패치·예외의 위치 고정

### 5.5.1 패치 허용 범위(결심층 한정)

방향 판단 프로세서에서 패치와 예외는 오직 기준5에서만 허용된다. 이는 규칙 증식이 계측이나 라벨 단계로 확산되는 것을 방지하기 위한 규정이다. 결심층에서의 패치는 “결심 허용 규정”의 수정이지, 계측 의미의 변경이 아니다.

### 5.5.2 규칙 비증식성 유지

패치는 누적되지 않는다<sup>8)</sup>. 반복 관측된 공통사항만이 규정으로 승격되며, 불규칙은 실패로 남겨 판례로 축적된다. 이 원리는 방향 판단이 규칙 덩어리로 붕괴되는 것을 방지한다.[14]

## 5.6 장 소결

본 장은 방향 판단 프로세서가 성능 이전에 구조적으로 닫힌 결심 경로를 갖고 있음을 설계와 연산 차원에서 제시하였다. 계측·라벨과 결심의 분리, 결심 레이어 단일화, 절차 레일 강제, 중단 규정은 방향 판단에서 조기결정과 사후변경을 구조적으로 불가능하게 만든다. 이 실증은 데이터 적중률을 주장하기 위한 것이 아니라, 결심 경로 누수가 발생하지 않는 실행 구조를 명세한 것이다. 다음 장에서는 동일한 닫힘 원리가 진동폭 처리 프로세서의 운용 사례에서 어떻게 유지되는지를 실증적으로 제시한다.

---

8) ‘규칙 비증식성’은 불규칙을 규칙으로 흡수해 덮지 않고, 공통사항만 최소 규정으로 승격하며 나머지는 실패(미스)로 보존하여 판례로 축적하는 원리이다. 패치는 결심층의 ‘허용 규정’에만 한정되며, 계측/라벨의 의미를 변경하는 형태로 확산되는 것을 금지한다.

## 제6장 실증 2: 진동폭 처리 프로세서의 운용 차원 구조적 닫힘

### 6.1 진동폭 처리 문제와 구조적 닫힘의 필요성

#### 6.1.1 수치 예측 영역에서의 결심 오염 위험

진동폭 처리는 예측 결과가 수치로 직접 노출되는 영역이다. 이로 인해 방향 판단보다 훨씬 강한 운영 압력을 받는다. 결과가 연속값으로 표현되기 때문에, 작은 오차는 “조정 가능한 편차”로 인식되기 쉽고, 사후 안정화(보정)가 개입할 여지가 커진다. 이 특성 때문에 진동폭 영역은 구조적 닫힘이 가장 먼저 붕괴되는 지점이 되기 쉽다.

실무적으로 진동폭 실패는 “틀렸다”보다 “조금 어긋났다”는 표현으로 처리되는 경우가 많다. 이 표현은 중립적으로 보이지만, 실제로는 결심 이후 결과 변경을 정당화하는 언어다. 본 연구는 이러한 관행이 예측 시스템을 연출 시스템으로 변질시키는 주요 원인이라고 본다.

#### 6.1.2 조기확정과 사후안정화의 결합

진동폭 처리에서 가장 위험한 실패 패턴은 조기확정과 사후안정화가 결합되는 경우다. 조기확정은 충분한 상태 분해 이전에 수치 범위가 사실상 확정되는 현상이며, 사후안정화는 그 결과를 목표 범위에 맞추어 조정하는 행위다. 이 두 과정이 결합되면 실패는 기록에서 완전히 사라진다.

이때 시스템은 “대체로 맞는 것처럼 보이는 상태”를 유지하지만, 실제로는 실패 원인을 축적하지 못한다. 진동폭 영역에서 구조적 닫힘이 요구되는 이유는 바로 이 지점이다. 결심 이전과 이후를 명확히 분리하지 않으면, 진동폭 예측은 운영자의 재량에 의해 무력화된다.

#### 6.1.3 운용 차원 닫힘의 목표

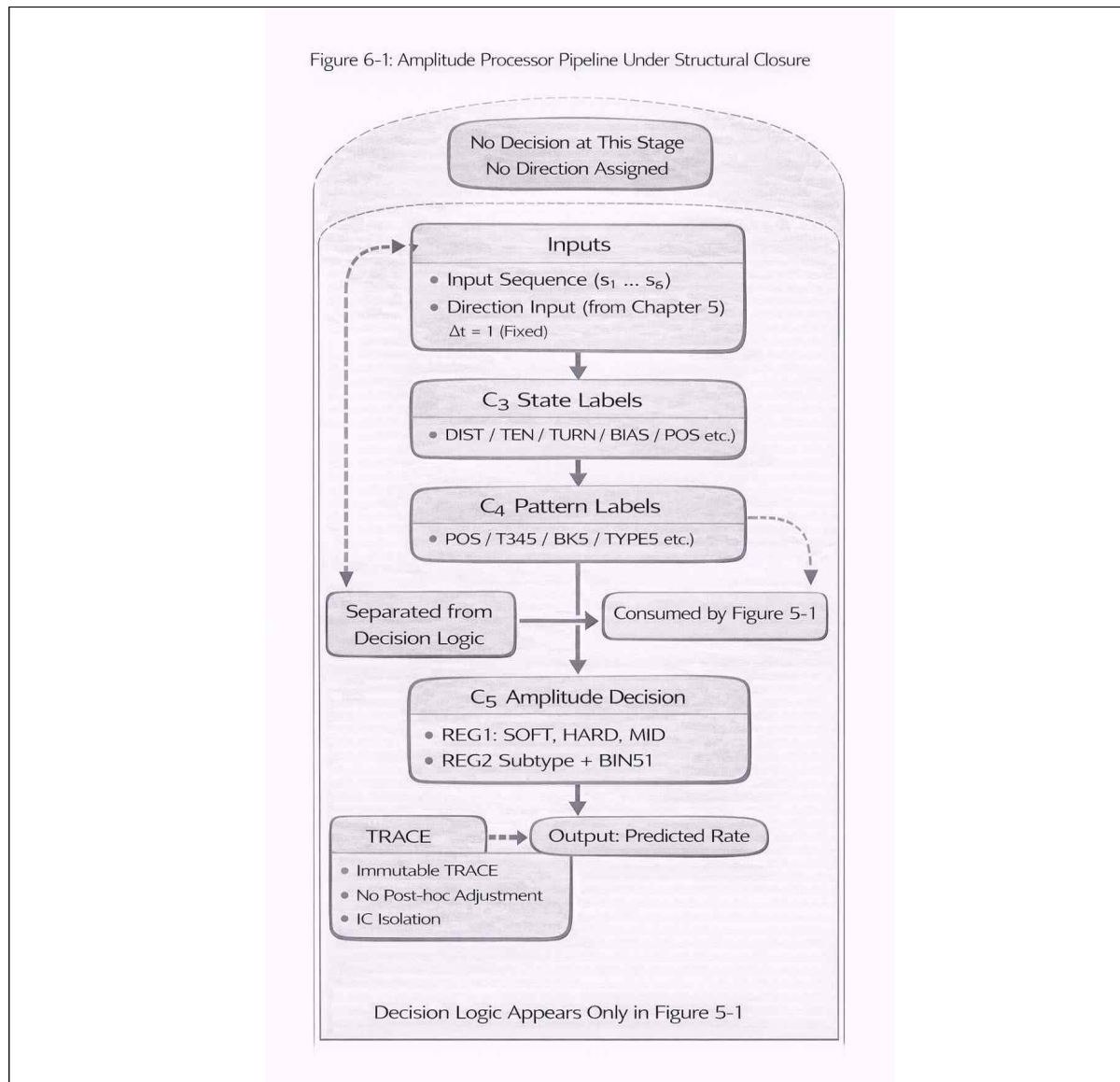
본 장의 목적은 진동폭 처리 프로세서가 운용 과정에서도 결심 경로를 봉인하고, 실패를 규칙으로 흡수하지 않으며, 실패를 판례로 축적하는 구조를 유지함을 입증하는 데 있다. 이는 성능 수치나 적중률을 주장하기 위한 실증이 아니라, 실패가 어떻게 처리되는지를 보여주는 실증이다.

### 6.2 진동폭 처리 파이프라인과 결심 경로 봉인

#### 6.2.1 파이프라인 개요

진동폭 처리 프로세서는 방향 판단 결과를 입력으로 받아, 이동 거리 기반의 진동폭 범주를 예측한다. 이 과정은 계측, 상태 분해, 패턴 라벨링, 결심 허용 판정, 출력의 순서로 구성되며, 각 단계는 명확히 분리된다.[9],[10]

특히 진동폭 처리에서는 기준2가 수치 생산자가 아니라 단계 분해 장치로 작동하며, 기준 3과 기준4가 상태와 패턴을 결합해 결심 입력을 형성한다. 결심은 오직 기준5에서만 발생하며, 이 구조는 방향 판단과 동일한 결심 레이어 단일화 원리를 따른다.

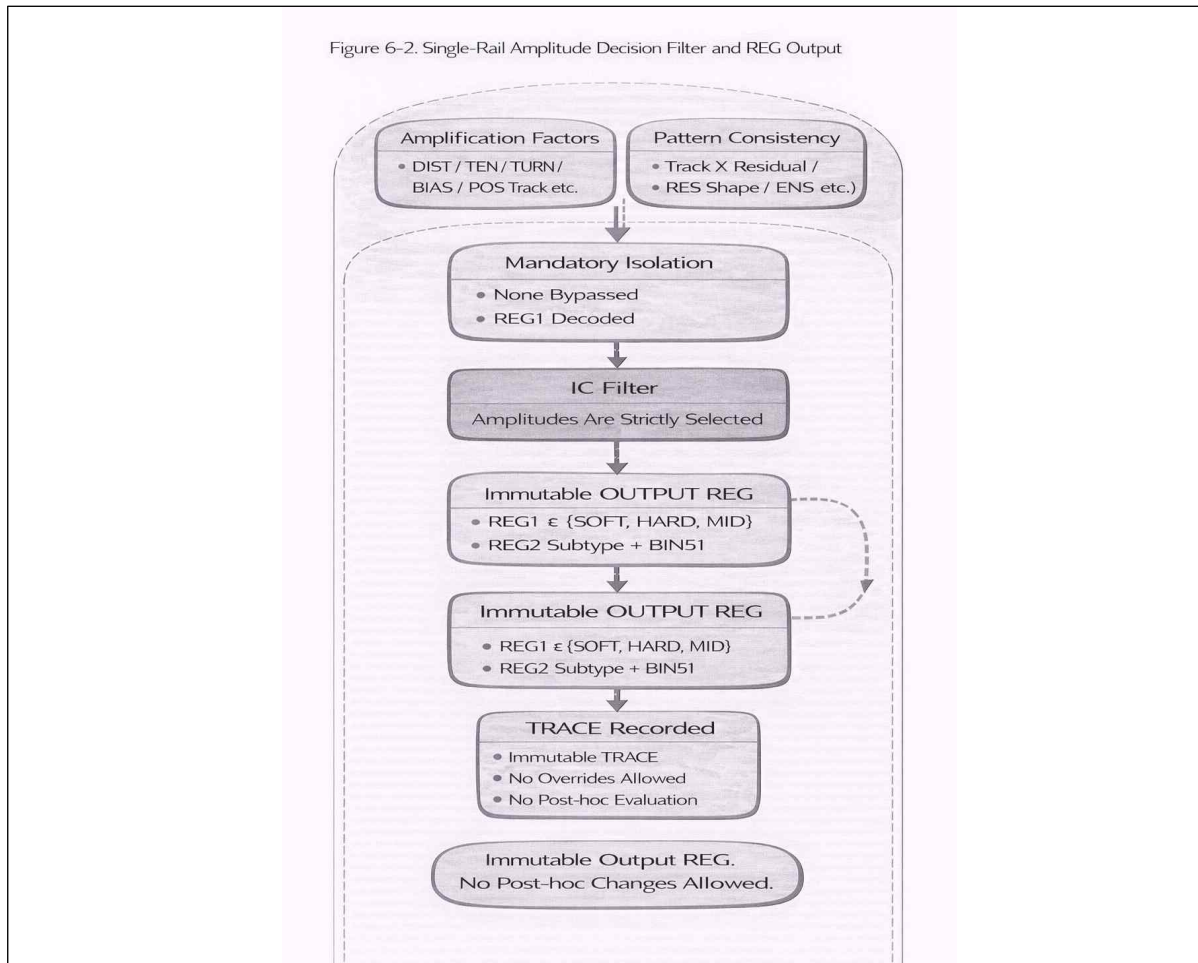


[그림 6-1. 구조적 닫힘 기반 진동폭 예측 파이프라인]

## 6.2.2 Mandatory Isolation과 IC Filter

진동폭 처리에서 Mandatory Isolation은 혼선 가능성이 높은 상태를 자동으로 격리하는 규정이다. 특정 조건에서는 결심이 즉시 허용되지 않고, 반드시 내부 확인(IC) 구간을 통과해야 한다. 이 규정은 조기확정을 구조적으로 차단한다.

IC Filter는 격리된 상태에 대해 추가 증거를 요구하며, 증거가 불충분할 경우 결심을 보류하거나 중단한다. 중요한 점은 IC가 "결론을 늦추는 장치"이지, 결론을 회피하는 장치가 아니라는 것이다. IC를 통과하지 못한 사례는 실패로 기록되며, 규칙으로 흡수되지 않는다<sup>9)</sup>.



[그림 6-2. 단일 레일 진동폭 결심 필터와 출력 봉인 구조]

## 6.3 실증 데이터의 지위와 한계

### 6.3.1 174 사례의 성격

본 연구에서 사용된 174 사례는 진동폭 처리 프로세서의 실제 운용 로그에서 수집되었다. 이 데이터는 성능 비교를 위한 벤치마크가 아니라, 구조적 단함이 운용에서 유지되는지를 관측하기 위한 사례 집합이다. 따라서 본 장은 통계적 유의성이나 일반화 가능성을 주장하지 않는다.

### 6.3.2 공개 범위와 해석 제한

해당 데이터는 비상업적 연구 목적에 한해 사용되며, 개별 사례의 원시 값은 공개되지 않는다. 본문에서는 표 형태의 요약 지표만을 사용한다. 이는 결과 재현이 아니라 구조 재현을 목표로 하기 때문이다. 다시 말해, 동일한 구조를 적용했을 때 동일한 종류의 실패가 어떻게 처리되는지를 보여주는 것이 목적이다.

9) 본 논문에서 실패(miss)는 제거되거나 은폐되어야 할 오류가 아니라, 이후 실행에서 예측 정확도를 향상시키기 위한 자산(asset)으로 취급된다. 실패 사례는 삭제·평활화되지 않고 판례(precedent)로 보존되며, 결심 경로를 수정하거나 규칙을 증식시키지 않은 상태에서 다음 실행의 판단 근거로 누적된다.

### 6.3.3 “하지 않은 주장”의 명시

본 장은 “진동폭을 잘 맞혔다”는 주장을 하지 않는다. 또한 “이 방법이 다른 데이터에서도 동일한 성능을 낼 것”이라는 일반화를 하지 않는다. 본 장의 주장은 오직 하나다. 진동폭 처리에서 결심 경로가 열려 있지 않으며, 실패가 규칙으로 흡수되지 않고 판례로 남는다는 점이다.

## 6.4 히트와 미스의 역할 분리

### 6.4.1 히트: 공통사항의 규칙화

진동폭 처리에서 히트는 “정답을 맞혔다”는 의미가 아니다. 히트는 공통 구조가 관측되었음을 의미한다. 즉, 특정 상태 조합과 패턴에서 반복적으로 유사한 결과가 관측될 경우, 이는 공통사항으로 분류된다. 공통사항은 규칙으로 정리될 수 있다.

그러나 이 규칙은 계측이나 라벨 의미를 변경하지 않는다. 규칙은 결심 허용 판단에만 제한적으로 사용된다. 이 제한이 없을 경우, 규칙은 곧바로 결심을 강제하는 수단으로 변질된다.

### 6.4.2 미스: 판례로서의 실패

미스는 실패이며, 동시에 자산이다<sup>10)</sup>. 미스는 규칙으로 흡수되지 않는다. 미스는 그대로 판례로 축적되며, 이후 유사한 구조가 재현되었을 때 참고 자료로만 사용된다. 이때 판례는 결론을 수정하는 수단이 아니라, 결심 안정성을 점검하는 입력으로 사용된다.

이 구조가 유지되기 때문에, 진동폭 처리 프로세서는 규칙 증식 없이도 점진적으로 안정화될 수 있다. 실패를 삭제하지 않기 때문에, 실패는 다음 실패를 예방하는 근거가 된다.

### 6.4.3 규칙 비증식성의 유지

히트와 미스의 분리는 규칙 비증식성 원리를 실현한다. 규칙은 공통사항만을 처리하며, 불규칙은 판례로 남는다. 이 분리가 무너지면, 시스템은 다시 규칙 증식의 길로 돌아가게 된다. 본 연구의 진동폭 처리 프로세서는 이 경계를 운용 차원에서 유지한다.

## 6.5 표 기반 운용 실증

### 6.5.1 표 구성 원칙

본 장에서 사용되는 표는 총 8개이며, 모두 본문에 직접 포함된다. 각 표는 특정 성능 수치를 강조하기 위한 것이 아니라, 결심 경로와 실패 처리 방식이 어떻게 작동했는지를 보여주기를 위한 것이다. 표는 결과 요약이 아니라 구조 관측 도구다.[9],[10]

10) 본 논문에서 ‘미스=자산’은 실패를 삭제하거나 보정으로 덮지 않고, 구조 좌표(상태·패턴·절차 경로)와 함께 기록하여 이후 유사 맥락에서 경고/위험도 판단의 근거로 소비한다는 의미이다. 즉, 실패를 ‘부끄러운 오류’가 아니라 시스템의 예측력을 높이는 판례 단위로 취급한다.

## 6.5.2 조기결정 차단 관측

[표 6-1. 진동폭 처리 단계별 역할과 결심 허용 여부]

단계	구성 요소	주요 역할	결정 권한	수정 허용 여부	비고
단계 1	계측 팩(Measurement Pack)	원시 시퀀스 및 파생 지표의 정량 계측	없음	불가	수치 신호만 생성하며 해석은 금지됨
단계 2	라벨 팩(Label Pack)	계측 신호의 구조적 라벨링	없음	불가	상태 토큰 및 구조 라벨 생성
단계 3	PCM (절차 제약 모델)	절차 게이트 제어 및 결심 허용성 판단	조건부	불가	결심 허용·차단 여부를 사전 결정
단계 4	IC (무결성 검사)	입력 완전성 및 내부 정합성 검증	거부권만	불가	필수 필드 누락 시 즉시 중단
단계 5	C5 (결심층)	최종 결심의 단일·비가역적 확정	전권	불가	단일 결심 지점, 되돌림 불가
단계 6	PVM (판례 검증 모듈)	과거 절차 추적과의 참조 비교	자문 전용	불가	결심 수정 없이 범위·정합성 보조

[표 6-2. 진동폭 예측값 분포 요약(174 사례)]

REG1 (최종)	사례수	AMP_pred 평균	AMP_pred 중앙값	최소	최대	25%	75%	IQR
SOFT	65	0.5396	0.5750	0.2250	1.1250	0.2250	0.5750	0.3500
MID	32	0.5172	0.5750	0.2250	1.1250	0.4250	0.5750	0.1500
HARD	77	1.4138	1.6750	0.4250	2.0000	1.1250	1.6750	0.5500

이 표들은 조기결정이 발생하지 않았음을 보여준다. 특정 조건에서 결심이 지연되거나 중단된 사례가 명시적으로 기록되며, 결심이 강제로 진행되지 않았음을 확인할 수 있다.[9],[10]

## 6.5.3 사후안정화 차단 관측

[표 6-3. 결심 경로 단일성 점검 결과]

점검 항목	정의	건수	비율
최종=판례경로 일치	REG_1C_FINAL = REG_1C_SIM	174	100.0%
최종=규칙경로 일치	REG_1C_FINAL = REG_1C_RULE	38	21.8%
규칙경로≠판례경로	REG_1C_RULE ≠ REG_1C_SIM	136	78.2%
전체		174	100.0%

[표 6-4. 진동폭 예측 실패 사례의 분포 요약]

오차 정의: ERR\_ABS

히트 기준: ERR\_ABS ≤ 0.10

오차 구간(ERR_ABS)	HARD	MID	SOFT	합계
0.10 이하	5	6	12	23
0.10~0.20	9	4	12	25
0.20~0.40	16	7	15	38
0.40~0.80	20	5	21	46
0.80~1.20	13	6	3	22
1.20 초과	14	4	2	20

이 표들은 출력 불변 규정이 운용에서 유지되었음을 보여준다. 예측 결과가 사후적으로 조정되지 않았으며, 실패가 실패로 남아 있음을 확인할 수 있다.[9],[10] 탐지·경고 계열의 신호는 결심을 수정하기 위한 수단이 아니라, 결심 레일을 침범하지 않는 범위에서 위험을 표시하는 입력으로만 사용되어야 한다 [15].

#### 6.5.4 판례 축적 관측

[표 6-5.경계 구간에서의 예측 오차 분포]

위치대 (POS_ZONE)	사례수	ERR_ABS 평균	ERR_ABS 중앙값	25%	75%	히트수 (ERR_ABS≤0.10)
BASE0	61	0.3609	0.2607	0.1162	0.5385	13
POS_MID	26	0.5426	0.4296	0.2200	0.7497	4
NEG_MID	23	0.4480	0.3639	0.1502	0.5236	3
POS_EDGE	28	0.7253	0.5860	0.3243	1.1301	2
NEG_EDGE	36	0.7164	0.6528	0.3214	1.0823	1

[표 6-6. 경계 및 중첩 구간의 세부 분포 요약]

위치대(POS_ZONE)	EDGE	TURN	합계
BASE0	0	61	61
POS_MID	0	26	26
NEG_MID	0	23	23
POS_EDGE	28	0	28
NEG_EDGE	36	0	36
합계	64	110	174



이 표들은 미스가 규칙으로 흡수되지 않고 판례로 축적되었음을 보여준다. 동일 유형의 실패가 “처리”되지 않고, 재등장 시 참고 대상으로 유지된다.[9],[10]

### 6.5.5 규칙 비증식성 관측

[표 6-7. 이웃 혼선 구간 관측 요약]

설명:

RECHECK\_USED = 재검사(격리 절차) 사용 여부(1/0)

CONF = 충돌/불확실 신호(1/0)

EXT\_FLAG = EXT 관련 플래그(1/0)

위치대 (POS_ZONE)	사례수	재검사 사용 (RECHECK_USED=1)	CONF 발생 (CONF=1)	EXT_FLAG=1
BASE0	61	43	5	12
POS_MID	26	21	5	4
NEG_MID	23	20	3	3
POS_EDGE	28	13	0	28
NEG_EDGE	36	11	0	36
합계	174	108	13	83

[표 6-8. 판례 기반 조회가 필요한 사례 분포]

DENY_TAG	의미(운용 정의)	사례수
NONE	차단 없음	167
DENY_HARD_CENTER	HARD 중심부 차단(예외 처리 구간으로 분리)	4
DENY_SOFT_EDGE	SOFT 경계 차단(예외 처리 구간으로 분리)	3
합계		174

이 표들은 규칙 수가 실패 수에 비례해 증가하지 않았음을 보여준다. 공통사항만이 규칙으로 정리되었고, 나머지는 판례로 유지되었다.[9],[10]

## 6.6 장 소결

본 장은 진동폭 처리 프로세서가 운용 과정에서도 구조적 닫힘을 유지함을 실증적으로 제시하였다. 결심 경로는 봉인되어 있었으며, 조기확정과 사후안정화는 구조적으로 차단되었다. 실패는 규칙으로 흡수되지 않았고, 판례로 축적되었다.

재현성 확보를 위해 본 논문은 입력 원시값의 전면 공개 대신, 운용 로그에서 생성된 온톨로지 좌표(onto\_tmp-5.csv)를 근거로 결심 레일의 분기(허용/격리/중단/경고)가 어떻게 기록되는지를 합성 TRACE 예시로 제시한다. 이 TRACE는 수치 재현이 아니라, PCM-IC-GroupHat-FINAL-PVM으로 이어지는 절차 재현을 목적으로 하며, 결심 커밋 이후 결론이 수정되지 않는다는 불변성과 무결성 위반 시 결심이 금지되는 Fail-Stop의 정상 동작을 확인하기 위한 최소 재현 단위다.

이 실증은 진동폭 예측의 정확도를 주장하기 위한 것이 아니다. 오히려 예측 시스템이 실패를 어떻게 다루는지가 시스템의 신뢰성과 지속 가능성을 결정한다는 점을 보여준다. 다음 장에서는 이러한 구조적 단함에 대해 제기될 수 있는 주요 반론들을 검토하고, 본 연구의 접근이 왜 구조적으로 타당한지를 논증한다.

## 제7장 논의: 예상 반론에 대한 구조적 봉인

### 7.1 논의의 기준과 범위

#### 7.1.1 본 장의 기준: 반론 대응이 아니라 전환의 증명

이 장은 반론을 하나씩 '설명으로 설득'하는 장이 아니다. 이 장의 목적은 구조적 닫힘이 기존의 성능 중심 접근을 보완하는 수준이 아니라, 연구의 기준선을 바꾸는 전환임을 증명하는 데 있다. 본 논문이 제시하는 전환은 세 가지다. 첫째, 규모의 경제에서 절차의 경제로 이동한다. 둘째, 결과를 사후에 다듬는 방식에서 결심 경로를 고정하는 방식으로 이동한다. 셋째, 수치 재현의 강박에서 절차 재현의 규격으로 이동한다.[1],[2],[5]

이 장의 반론들은 모두 이 세 전환이 왜 필요하고 피할 수 없는지 드러내는 장치로 배치된다.

#### 7.1.2 논의 범위: 성능 대신 '운용 가능한 책임'을 다룬다

본 논문은 모델이 얼마나 맞는지 경쟁하지 않는다. 성능은 개선될 수 있으나, 결심 경로가 열려 있으면 운용 환경에서 결과는 쉽게 오염된다. 사후 보정이 개입하고, 그 순간 무엇이 결심이었는지, 왜 결심이 되었는지, 누가 책임을 지는지가 사라진다.[1],[2],[5]

따라서 본 장은 성능 중심의 논의를 벗어나, 결심이 허용되는 조건과 결심이 금지되는 조건, 그리고 결심이 남기는 기록이 어떻게 책임과 재현성을 구성하는지에 집중한다. 이것이 절차의 경제가 요구하는 기준이다.

### 7.2 반론 1: “구조적 닫힘은 개념적 말장난이다”

구조적 닫힘을 추상적 표어로 보는 반론은 가장 먼저 등장한다. 이 반론은 구조적 닫힘이 기술이 아니라 담론이라고 주장한다.

#### 7.2.1 말이 아니라 '닫히는 것'이 무엇인지 먼저 본다

구조적 닫힘이 닫는 것은 정보가 아니다. 구조적 닫힘이 닫는 것은 결심이 발생할 수 있는 통로다. 운용에서 문제가 되는 것은 부족한 지식이 아니라, 공백이 결론으로 승격되는 순간이다. 구조적 닫힘은 이 승격을 허용하지 않는다. 결심이 발생하는 절차를 고정하고, 절차를 벗어나면 결론이 아니라 중단이 남는다.

이때 닫힘은 설명이 아니라 구현 조건이 된다. 그림 4-6이 권한 경계를, 그림 4-7이 기록 단위를 보여주는 이유도 여기에 있다. 닫힘은 “그럴듯한 표현”이 아니라 “허용/금지의 실행 규격”으로 존재한다.

### 7.3 반론 2: “단일 규칙이 아니라 더 많은 규칙이 필요하다”

두 번째 반론은 개선의 방식에 관한 것이다. 실패가 나오면 규칙을 추가해 덮어야 한다는 주장이다.

#### 7.3.1 이 반론은 ‘결과 중심’ 사고의 전형이다

더 많은 규칙은 결과를 더 잘 통제하는 것처럼 보인다. 그러나 규칙이 늘어날수록 결심은 단일 사건이 아니라 다수 규칙의 합성 결과가 된다. 그 순간 결심 경로는 다중화되고, 실패는 규칙 속으로 묻힌다. 결과는 그럴듯해질지 모르지만, 운용자는 왜 그런 결론이 나왔는지 설명할 수 없게 된다.[3],[14]

구조적 단점은 규칙의 양이 아니라 규칙의 위치를 고정한다. 규칙은 상태를 만들 수는 있지만, 결심을 만들 수는 없다. 결심은 상위층의 단일 사건으로만 허용된다.

#### 7.3.2 두 번째 전환: ‘사후 보정’이 아니라 ‘실패를 남기는 개선’으로 이동한다

이 논문에서 실패는 제거 대상이 아니다. 실패는 개선의 근거다. 그래서 실패가 발생하면 규칙이 늘어나는 것이 아니라, 판례가 축적된다. 이 축적은 결심 경로를 흔들지 않도록 결심 이후에만 제한적으로 참조된다.[3],[14]

여기서 두 번째 전환이 성립한다. 결과를 수정해 실패를 숨기는 방식에서, 결심 경로를 고정하고 실패를 기록으로 남기는 방식으로 이동한다.

### 7.4 반론 3: “KG 스코프 제한이면 충분하다”

세 번째 반론은 지식의 범위를 좁히면 문제는 사라진다는 주장이다. 즉, 참조를 제한하면 환각이 줄고 결심도 안정될 것이라는 기대다.

#### 7.4.1 KG는 ‘무엇을 아는가’를 줄이지만 ‘어떻게 결심하는가’를 고정하지 못한다

KG는 유용하다. 하지만 KG가 해결하는 것은 지식 오염의 일부다. 결심 경로가 열려 있으면 KG 안에서도 결심은 사후적으로 수정될 수 있고, 절차는 흔들릴 수 있다.

구조적 단점의 관심사는 지식의 범위가 아니라 결심의 형식이다. KG는 보조 도구가 될 수 있으나, 결심 경로 봉인을 대체할 수는 없다.

#### 7.4.2 이 반론은 ‘지식 통제’로 ‘책임’을 대체하려는 시도다

KG로 지식을 통제하면 결과는 안정돼 보일 수 있다. 그러나 책임 가능한 결심은 지식의 제한이 아니라 절차의 고정에서 나온다. 지식이 아니라 결심 경로를 봉인해야 운용 책임이 생긴다. 이 장의 전환 논리는 여기서 더욱 분명해진다. 절차가 핵심이고, 지식은 그 절차 안에서만 의미를 가진다.

## 7.5 반론 4: “중간층·상위층 구현은 과도하다”

네 번째 ‘구현 과도’라는 프레임으로 구조를 흔드는 시도다.

### 7.5.1 단일 층 해법은 구현이 아니라 책임을 무너뜨린다

단일 층은 단순해 보인다. 그러나 단순해 보이는 만큼 결심은 내부에 숨는다. 계측과 라벨과 결심이 섞이면 결론은 나오지만, 왜 나왔는지 남지 않는다. 이 구조에서는 실패가 나오면 규칙을 더 붙이거나 사후적으로 다듬는 길밖에 없다. 결과 중심의 악순환이 재도입된다.

### 7.5.2 절차의 경제는 계층 분리를 요구한다

절차의 경제에서 중요한 것은 구현 단순성이 아니라 결심 단일성이다. 중간층은 판단하지 않고 상태를 제공하고, 상위층은 그 상태를 받아 결심을 허용하거나 금지한다. 이 분리가 있어야 결심은 명시적 사건이 되고, 커밋 불변과 Fail-Stop이 의미를 가진다. 즉, 계층 분리는 과도가 아니라 전환의 조건이다.

## 7.6 반론 5: “PVM은 사후 보정과 다르지 않다”

마지막 반론은 판례 참조가 결국 결론 수정으로 흐를 것이라는 의심이다.

### 7.6.1 PVM이 작동하는 순간은 ‘결심 이후’로 고정된다

사후 보정은 결심 이전/이후를 가리지 않고 결과를 바꾼다. 그러나 PVM은 결심 이전에 개입하지 않는다. 결심은 절차 레일을 통과해 커밋된 사건이고, PVM은 그 이후에만 경고와 위험 신호를 제공한다. 이 시점 제한이 사후 보정과의 경계다.[9],[10],[14]

### 7.6.2 세 번째 전환: 재현을 ‘숫자’가 아니라 ‘절차’로 정의한다

사후 보정은 결과를 바꾼다. 그러면 재현은 깨진다. 왜냐하면 무엇이 결심이었는지 남지 않기 때문이다. 구조적 닫힘은 결심 커밋 이후 결론을 수정하지 않는다.[13] 대신 TRACE를 남기고, 그 TRACE는 온톨로지 좌표로 축적된다. 그리고 부록 G는 그 구조를 절차 재현 단위로 제시한다.[9],[10],[14]

여기서 세 번째 전환이 완성된다. 재현의 대상은 수치가 아니라 결심 경로다. 이것이 운용 가능한 재현성이다.

## 7.7 장 소결

본 장의 반론들은 모두 하나의 방향으로 수렴했다. 규모를 키우거나 규칙을 늘리거나 지식을 제한해 결과를 안정시키려는 시도는 결국 결심 경로를 열어 둔 채 결과만 다듬는 방식으로 귀결된다. 그 방식은 단기적으로는 그럴듯하지만, 운용 환경에서는 책임성과 재현성을 붕괴시킨다.

본 논문이 제시하는 전환은 보완이 아니라 전복이다. 규모의 경제가 아니라 절차의 경제로 이동한다. 결과 중심이 아니라 결심 경로 중심으로 이동한다. 수치 재현이 아니라 절차 재현으로 이동한다. 구조적 닫힘은 이 세 전환을 실행 규격으로 고정하며, 그것이 세계에 제시하는 새로운 기준선이다.

다음 장에서는 본 논문의 기여를 정리하고, 구조적 닫힘이 예측 시스템 전반에 갖는 의미와 향후 연구 과제를 제시한다.

## 제8장 결론

### 8.1 연구 요약

#### 8.1.1 문제의 재정의

본 연구는 LLM 기반 예측 시스템의 실패를 성능 부족, 환각, 규칙 미비로 환원하는 관행을 비판하고, 실패의 근본 원인을 '결심 경로 문제'로 재정의하였다. 즉, 오류는 중간 산출물의 생성 그 자체에서 발생하는 것이 아니라, 검증되지 않은 산출물이 결론으로 승격될 수 있는 경로가 구조적으로 열려 있을 때 반복된다. 성능이 향상될수록 실패가 교묘해지고 은폐되는 현상은 이 재정의에 의해 설명 가능해진다.

#### 8.1.2 해결의 방향: 결심 경로 봉인

이 재정의에 따라 해결의 방향은 명확해진다. 결심이 발생할 수 있는 위치를 단일화하고, 결심이 허용되기 위한 최소 조건과 절차 레일을 강제하며, 결심 이후 결과 변경을 금지하고, 실패를 규칙으로 흡수하지 않고 판례로 축적해야 한다. 본 연구는 이러한 실행 규칙의 집합을 '구조적 단함'이라 정의하였다.

### 8.2 핵심 기여

#### 8.2.1 구조적 단함의 실행 규칙화

본 연구의 첫 번째 기여는 구조적 단함을 철학적 은유가 아니라 실행 규칙으로 정식화한 데 있다. 구조적 단함은 조기결정 금지, 사후변경 금지, 결심 레이어 단일화, 절차 레일 강제, 누락·불일치 중단 규정으로 구성되며, 이 규정들은 위반 시 중단되는 형태로 구현될 수 있다. 이를 통해 결심은 "자연스럽게 발생하는 출력"이 아니라 "허용되는 전이"로 전환된다.

#### 8.2.2 NCAF·PCM·PVM의 역할 분리와 통합 구조 제시

두 번째 기여는 NCAF·PCM·PVM을 결심 경로 봉인을 위한 코어 기술로 명세하고 통합 실행 구조를 제시한 데 있다. NCAF는 레이어 권한 체계를 제공하여 계층·라벨이 결론을 침범하지 못하게 하고, PCM은 결심 허용 최소 조건과 절차 레일 강제로 결심 오염을 차단한다. PVM은 PCM 이후에만 제한적으로 작동하며, 실패를 규칙이 아니라 판례로 축적함으로써 규칙 비증식성을 유지한다. 이 구조는 단일 모듈의 고도화가 아니라, 결심 책임을 단일 지점에 귀속시키는 실행 구조라는 점에서 차별적이다.

#### 8.2.3 실증 1: 방향 판단 프로세서의 설계·연산 차원 단함 제시

세 번째 기여는 방향 판단 프로세서가 데이터 적중률 이전에 설계·연산 차원에서 구조적

으로 닫힌 결심 경로를 갖고 있음을 제시한 데 있다. 계측·라벨과 결심의 분리, 결심층의 유일성, 절차 레일 고정, 중단 규정은 결심 경로 누수를 차단하며, 이는 성능 논의와 독립적인 구조적 안전성으로 해석된다.

#### 8.2.4 실증 2: 진동폭 처리 프로세서의 운용 차원 닫힘 입증

네 번째 기여는 진동폭 처리 프로세서의 운용 로그(174 사례)를 통해 구조적 닫힘이 운용 과정에서도 유지됨을 실증적으로 제시한 데 있다. 진동폭 영역은 수치 예측이 직접 노출되기 때문에 조기확정과 사후안정화의 유혹이 가장 강한 구간이다. 그럼에도 실패는 규칙으로 흡수되지 않고 판례로 축적되며, 공통사항만이 제한적으로 규칙화되는 방식이 유지된다. 이는 구조적 닫힘이 선언이 아니라 운용을 건디는 실행 구조임을 시사한다.

### 8.3 해석과 함의

#### 8.3.1 “정확도”보다 “책임성과 재현성”의 우선

본 연구는 예측 시스템의 핵심 품질을 정확도 단일 지표로 보지 않는다. 정확도는 중요하지만, 결심 경로가 열려 있는 시스템에서 정확도는 쉽게 연출될 수 있으며, 실패의 원인을 삭제한 채 개선된 것처럼 보일 수 있다. 반면 구조적 닫힘은 결심의 위치와 절차를 고정함으로써 책임성과 재현성을 확보한다. 이는 예측 시스템이 실무에서 신뢰되기 위한 최소 조건이며, 성능 향상과 별개로 요구되는 차원의 품질이다.

#### 8.3.2 규칙 증식의 유혹과 규칙 비증식성의 필요

실무 환경에서 규칙 증식은 가장 손쉬운 대응이지만, 장기적으로 경계 붕괴와 충돌을 낳고 실패를 삭제한다. 본 연구가 제시한 규칙 비증식성 원리는 규칙을 금지하는 것이 아니라, 규칙이 개입할 수 있는 위치를 결심층으로 고정하고, 공통사항에 한해 제한적으로 승격시키며, 불규칙은 판례로 남겨 운용 안정성을 높이는 방향을 제시한다. 이는 불규칙을 규칙으로 통제하려는 시도를 포기하는 것이 아니라, 불규칙을 실패로서 보존함으로써 시스템이 개선될 수 있게 만드는 운영 원리다.

#### 8.3.3 KG 스코프 제한과의 관계

지식 그래프 기반 스코프 제한은 임의의 연결 생성과 무분별한 추론을 억제하는 데 유효하다. 그러나 스코프 제한은 결심의 발생 조건과 절차를 자동으로 고정하지 않는다. 본 연구의 결론은 KG를 대체하는 것이 아니라, KG가 실무에서 책임성과 재현성으로 이어지기 위해서는 결심 경로 봉인이 선행되어야 한다는 점이다. 즉, 구조적 닫힘은 스코프 제한과 경쟁하는 패러다임이 아니라, 스코프 제한의 효과를 “결심 구조”로 연결시키는 상위 규격으로 해석될 수 있다.[1],[2]



## 8.4 한계

### 8.4.1 실증 범위의 제한

본 연구의 실증은 방향 판단 프로세서의 구조 명세와 진동폭 처리 프로세서의 운용 로그 (174 사례)에 기반한다. 본 논문은 이 데이터로 일반적 성능 우위를 주장하지 않으며, 다른 도메인이나 다른 데이터 분포에서 동일한 결과가 자동으로 재현될 것이라고 단정하지 않는다. 본 연구의 실증은 '정답률 입증'이 아니라 '결심 경로 봉인 유지'의 관측이다.

### 8.4.2 판례 활용의 운영 설계 과제

PVM은 결론을 수정하지 않으면서 결심 안정성을 점검하는 것을 목표로 하지만, 실제 운용에서 판례 참조가 사용자에게 '사후 보정'으로 오해될 가능성은 남아 있다. 이를 방지하기 위해 판례 참조의 시점, 출력 형식, 경고 방식, 기록 구조를 더 명확히 표준화할 필요가 있다. 즉, 판례의 "참조"와 결과의 "수정"을 명확히 분리하는 운영 규격이 추가적으로 요구된다.[1],[2]

## 8.5 향후 연구

### 8.5.1 결심 경로 봉인의 일반화

향후 연구는 구조적 단힘의 규격을 다른 유형의 예측·분류·의사결정 시스템에 확장하는 방향으로 진행될 수 있다. 특히 결심 레이어 단일화와 절차 레일 강제는 도메인 특화 규칙과 무관하게 적용 가능한 상위 원리로 간주될 수 있으며, 이를 일반화한 형식 명세가 필요하다.

### 8.5.2 실패의 표준 좌표화와 판례 상호운용

판례 축적이 효과적으로 작동하기 위해서는 실패를 표준 좌표로 표현하는 방식이 요구된다. 서로 다른 시스템이나 서로 다른 환경에서 발생한 실패가 동일한 구조 좌표로 매핑될 수 있다면, 판례는 개인 시스템의 내부 자산을 넘어 상호운용 가능한 지식 기반으로 확장될 수 있다.

### 8.5.3 결심 책임의 검증 가능성

결심 경로가 봉인된 시스템은 책임성을 강화하지만, 그 책임성이 외부 검증 가능성으로 이어지기 위해서는 결심 레일과 로그 구조의 공통 표준이 필요하다. 향후 연구는 결심 레일의 표준화와 외부 감사 가능성(auditability)을 결합하는 방향으로 확장될 수 있다.

## 8.6 결론

본 연구는 LLM 기반 예측 시스템의 실패를 '결심 경로 문제'로 재정의하고, 결심이 발생

할 수 있는 경로를 구조적으로 봉인하는 실행 규격으로서 '구조적 닫힘'을 제시하였다. NCAF·PCM·PVM의 역할 분리와 통합 실행 구조는 결심 책임을 단일 레이어에 귀속시키고, 조기결정과 사후변경을 구조적으로 차단하며, 실패를 규칙이 아닌 판례로 축적하게 한다. 방향 판단 프로세서는 설계·연산 차원에서, 진동폭 처리 프로세서는 운용 차원에서 이 원리가 유지됨을 제시하였다.

결국 예측 시스템의 신뢰성은 더 많은 규칙이나 더 큰 모델만으로 확보되지 않는다. 신뢰성은 결심이 어디서, 어떤 절차로 허용되는지에 대한 실행 구조의 문제이며, 구조적 닫힘은 그 문제에 대한 실무적 해법으로 제안된다.

**본 연구는 비상업적 연구 및 학술 목적 공개를 전제로 한다.**

**본 자료의 전부 또는 일부를 시스템에 적용하거나 실전 의사결정에 사용하는 행위는 사전 협의 및 명시적 허가 없이는 허용되지 않는다.**

## 참고문헌

1. National Institute of Standards and Technology (NIST). (2024).  
Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). NIST.
2. National Institute of Standards and Technology (NIST). (2023).  
Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST AI 100-1). NIST.
3. OWASP. (2024).  
OWASP Top 10 for Large Language Model Applications. OWASP Foundation.
4. OpenAI. (2025).  
Model Spec (2025-04-11). OpenAI.
5. Anthropic. (2024).  
Responsible Scaling Policy (RSP). Anthropic.
6. Islam Tonmoy, M. T., et al. (2024).  
A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. arXiv:2401.01313.
7. Wang, L., et al. (2024).  
A Survey on Large Language Model Based Autonomous Agents. Frontiers of Computer Science.
8. Guo, T., et al. (2024).  
Large Language Model Based Multi-Agents: A Survey of Progress and Challenges. IJCAI 2024 Proceedings.
9. Es, S., et al. (2023).  
RAGAS: Automated Evaluation of Retrieval Augmented Generation. arXiv:2309.15217.
10. Es, S., et al. (2024).  
RAGAs: Automated Evaluation of Retrieval Augmented Generation. EACL 2024 (Demo).

11. Yao, S., et al. (2022).

ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

12. Schick, T., et al. (2023).

Toolformer: Language Models Can Teach Themselves to Use Tools. NeurIPS 2023.

13. Madaan, A., et al. (2023).

Self-Refine: Iterative Refinement with Self-Feedback. NeurIPS 2023.

14. Dong, Y., et al. (2024).

Safeguarding Large Language Models: A Survey. arXiv:2406.02622.

15. NIST. (2025).

Hallucination Detection in Large Language Models Using Diversion Decoding. NIST CSRC.

**본 논문은 비상업적 연구 및 학술 목적에 한하여 공개된다.**

**단순 열람 및 인용을 넘어서는 모든 활용(재배포, 강의자료화, 시스템/모델 적용, 실전 의사결정 적용 등)은**

**사전에 저자와 협의하여 명시적 허가를 받아야 한다.**

**본 논문의 전부 또는 일부에 대한 상업적 사용은 엄격히 금지된다.**

**본 논문은 동일 연구 내용을 담은 국문/영문 논문을 병렬로 공개한 것이며, 모든 언어 버전에 동일한 사용 조건과 라이선스가 적용된다.**

## 부 록

### 부록 A. 용어 및 약어

A-1. 구조적 닫힘(Structural Closure): 결심이 발생할 수 있는 경로를 봉인하는 실행 규칙의 집합(조기결정 금지, 사후변경 금지, 결심 레이어 단일화, 절차 레일 강제, 중단 규정).

A-2. 결심(Decision/Commit): 결론이 출력으로 확정되는 단일 사건. 결심 이전 산출물은 결론을 강제할 수 없다.

A-3. 결심 경로(Decision Path): 입력→계측→라벨→(격리/검사)→결심→출력으로 이어지는 단일 레일 절차.

A-4. 공백 봉인(Blank Sealing): 미정/불확실이 존재해도 그것이 결론으로 승격되는 통로를 차단하는 규정.

A-5. NCAF: 레이어 권한 분리 프레임(계측/라벨/결심의 역할·금지 고정).

A-6. PCM: 결심 허용 최소 조건 + 절차 레일 강제 + 위반 시 중단 규정.

A-7. PVM: PCM 이후에만 제한적으로 판례를 참조하여 결심 안정성을 점검(결론 수정 금지).

A-8. TRACE: 결심 이전/이후의 실행 흔적을 불변으로 기록하는 구조.

A-9. 규칙 비증식성(Non-Proliferation of Rules): 불규칙을 규칙으로 흡수하지 않고 판례로 축적하며, 공통사항만 제한적으로 규칙화.

**본 부록은 본문과 동일한 비상업/사전협의 조건을 따른다**

## **부록 B. 실행 규격(TRACE)**

### **B-1. TRACE 필수 필드(최소)**

1. 입력 시퀀스( $s_1 \sim s_5$ ),  $\Delta t$ , (방향/진동폭이라면) 모듈명/버전
2. 계측 레이어 산출물(C2)
3. 상태 라벨(C3) + 패턴 라벨(C4)
4. PCM 판정(허용/격리/중단) + 사유 코드
5. 결심층 산출물(C5: FINAL)
6. 출력 불변 서명(해시/커밋/런ID 중 택1 이상)

**B-2. 불변 규정: 결심 이후 값은 수정·덮어쓰기 금지. 재실행은 “새 TRACE”로만 남김.**

**B-3. 중단 기록 규정: 누락/불일치/순서 위반 시 결론 대신 중단 코드와 함께 TRACE 종료.**

## 부록 C. NCAF 규정(레이어 권한/금지)

C-1. 계측 레이어 금지: 결론(방향/진동폭)을 직접·간접으로 판정/암시하는 산출 금지.

C-2. 라벨 레이어 금지: 라벨이 결론 토큰으로 변질되는 모든 규칙(“이 라벨이면 결론 =...” 형태) 금지.

C-3. 결심 레이어 금지: 계측 정의/라벨 정의를 사후적으로 변경하는 행위 금지.

### C-4. 침범 판정:

1. 결론이 중간층에서 사실상 확정됨,
2. 결심층이 중간층 의미를 바꿈,
3. 절차 생략 — 발생 시 구조 위반.

#### 부록 D. PCM 규정(결심 허용 최소 조건/레일/중단)

D-1. Minimum Admissibility(형식 조건): 필수 계측·필수 라벨 누락 0, 순서 위반 0, 불일치 탐지 시 중단.

D-2. 내용 조건: 혼선 가능 신호가 존재하면 결심 즉시 허용 금지 → 격리(IC) 절차로 이동.

D-3. Sequence Lock(레일 강제): (예) GATE\_PRE → IC → GroupHat → FINAL 같은 고정 순서 외 진행 금지.

D-4. Fail-Stop: 위반 탐지 시 “결론 대신” 중단 기록으로 종료(실패를 삭제하지 않음).



부록 E. PVM 규정(판례 참조/규칙 비증식성)

E-1. 작동 시점 제한: PCM이 '허용'한 이후에만 판례 조회 가능.

E-2. 결론 수정 금지: 판례는 결론을 바꾸지 않으며, 위험도 표기/경고/범위 조정 같은 "안정성 점검"만 수행.

E-3. 축적 단위: 실패는 "수정 대상"이 아니라 "판례 레코드"로 저장(동일 구조 재출현 시 참조).

E-4. 규칙 비증식성: 공통사항만 규칙화(결심층 내부). 불규칙은 규칙으로 흡수 금지.

## **부록 F. 그림·표 목록**

### **F-1. 그림 목록(15)**

1. 그림 2-1. 스케일 아웃 접근과 구조 중심 접근의 대비
2. 그림 3-1. Open-world 추론과 구조적 닫힘(Structural Closure)의 대비
3. 그림 3-2. 단계적 전이와 결심 경로 봉인 개념도
4. 그림 3-3. 결심 경로 단일화와 공백 봉인의 전체 흐름
5. 그림 4-1. NCAF·PCM·PVM 코어 구조 개요
6. 그림 4-2. NCAF(상위 제약/정렬 프레임)의 역할
7. 그림 4-3. PCM(절차 레일)의 핵심 동작(허용/격리/중단)
8. 그림 4-4. PVM(판례 참조)의 작동 범위(PCM 이후, 결론 수정 금지)
9. 그림 4-5. 단일 결심 레이어와 커밋(Commit) 불변 구조
10. 그림 4-6. 권한 매트릭스(Authority Matrix): NCAF·PCM·PVM·C5(Commit)·TRACE의 허용/금지 경계
11. 그림 4-7. TRACE-온톨로지 스키마(TRACE/Ontology Schema): 절차 재현 단위의 기록 구조
12. 그림 5-1. 방향 결심(25/75) 처리 구조
13. 그림 5-2. 상태 표현 계층과 라벨/토큰 구조
14. 그림 6-1. 진동폭 처리 프로세서(1차 분류/절차 레일)
15. 그림 6-2. 진동폭 출력(서브유형/BIN)과 커밋 불변 구조

### **F-2. 표 목록(8)**

1. 표 6-1. 표 구성 원칙 및 필드 커버리지
2. 표 6-2. 관측 로그: 조기 결심 차단
3. 표 6-3. 관측 로그: 사후 안정화(수정) 차단
4. 표 6-4. 관측 로그: 격리(ISOLATE) 이벤트 및 IC 결과
5. 표 6-5. 관측 로그: 판례 축적(PVM) 엔트리
6. 표 6-6. 관측 로그: 규칙 비증식(비증식성) 점검
7. 표 6-7. 대표 사례 요약: REGIME-결과 정합
8. 표 6-8. 대표 사례 요약: 실패=자산 추적성(TRACE)

## 부록 G. 합성 TRACE 3케이스(절차 재현용)

본 부록은 운용 로그 온톨로지(onto\_tmp-5.csv)에 기록되는 필드 구조(예: COORD\_VER, REG\_DECISION\_PATH, RISK\_TAG, CAUSE\_CODE, DEC\_CODE, REG\_1C\_\*, C5\_PATH 등)를 기준으로, 수치 재현이 아니라 절차 재현을 위해 구성한 합성 TRACE 예시이다. 아래 3케이스는 PCM-IC-GroupHat-FINAL-PVM 레일에서 대표적으로 발생하는 분기(확인 커밋 / 중단 / 경고 동반 커밋)를 각각 1개씩 제시한다. 본 예시는 입력 원시값(s1~s5) 재현을 포함하지 않는다.

### G.1 CASE A — 확인 커밋(RULE→SIM\_CONFIRM)

이 케이스는 규칙 판단 이후 확인 단계(검증 강제 구간)를 거쳐 결심이 커밋되는 흐름을 나타낸다. 로그 필드 관점에서 REG\_DECISION\_PATH=RULE→SIM\_CONFIRM로 요약되며, 약한 위험 신호가 동반될 수 있다(예: RISK\_TAG=WEAK\_EXT\_TRAP). 원인 토큰은 "규칙 단독으로는 케이스 분류가 불충분함"을 나타내는 형태로 기록될 수 있다

(예: CAUSE\_CODE=REG\_FAIL\_NOCASE|WEAK\_EXT\_TRAP). 이 경우에도 절차 레일은 이탈하지 않으며, 확인 단계를 통과하면 최종 결심은 커밋된다. 핵심은 "위험 신호가 있다고 해서 사후 수정으로 넘어가는 것이 아니라, 확인 단계가 레일 내부에서 작동하고 커밋으로 종결된다"는 점이다.

### G.2 CASE B — Fail-Stop(결론 미산출)

이 케이스는 무결성 위반(누락/불일치/순서 위반)이 탐지될 때 PCM이 결론 산출을 허용하지 않고 즉시 중단하는 정상 동작을 나타낸다. 로그에서는 REG\_DECISION\_PATH=STOP 또는 이에 준하는 중단 경로로 요약되며, 중단 사유는 누락/불일치 계열 코드로 기록된다(예: CAUSE\_CODE=INTEGRITY\_FAIL|MISSING\_REQUIRED). 이 경우 FINAL은 산출되지 않는다. 중요한 점은 이것이 "보류"나 "회피"가 아니라, 결심 경로 오염을 방지하기 위한 결론 금지(Fail-Stop)라는 규격적 동작이라는 점이다. 즉, 시스템은 실패를 지우지 않고 STOP\_CODE 및 TRACE로 남긴다.

### G.3 CASE C — 경고 동반 커밋(RULE→SIM\_OVERRIDE)

이 케이스는 결심이 커밋된 이후, PVM이 위험 신호를 경고로 기록하되 결론을 변경하지 않는 흐름을 나타낸다. 로그 관점에서 REG\_DECISION\_PATH=RULE→SIM\_OVERRIDE와 같은 형태로 요약되며, 위험 신호는 RISK\_TAG로 기록될 수 있다

(예: RISK\_TAG=TURN\_TRAP). 원인 토큰은 규칙 기반 분류의 불확실성을 나타내는 코드와

위험 태그가 결합된 형태로 기록될 수 있다

(예: CAUSE\_CODE=REG\_FAIL\_NOCASE|TURN\_TRAP). 이 케이스에서 PVM은 경고/위험도/예측 범위 조정과 같은 메타 신호를 제공하지만, FINAL은 커밋 이후 불변이며 수정되지 않는다. 이로써 PVM이 “사후 보정 도구”가 아니라 “커밋된 결심의 안정성을 점검하고 운용상 위험을 표기하는 장치”임이 확인된다.

#### **G.4 요약**

CASE A는 확인 단계 후 커밋(레일 유지), CASE B는 무결성 위반 시 결론 금지(Fail-Stop), CASE C는 커밋 이후 경고만 제공(PVM)이며 결론 수정은 발생하지 않는다. 이 3케이스는 원시 입력값을 재현하지 않더라도, 온톨로지 로그 필드 구조를 통해 결심 경로의 분기와 봉인 규칙(레일 강제, Fail-Stop, 커밋 불변)을 절차 수준에서 재현하는 최소 단위로 기능한다.