

Same Answers, Different Active-Store States: An Identification Audit of FactConsolidation Single-Hop

Daeun Jeong Yongjun Zhu

Yonsei University

da3967@yonsei.ac.kr, zhu@yonsei.ac.kr

Abstract

After a factual conflict, the same credited answer can arise from different record-choice rules and different active-store states. We audit this ambiguity in MemoryAgentBench’s 400-question FactConsolidation Single-Hop (FC-SH) task. A fixed exact-template parser identifies 282 questions with two target arrivals and distinct values, plus 118 singletons. In every two-arrival case, the larger serial also arrives last, so last-arrival and maximum-serial selection coincide. We cross these selectors with stores that expose either one active record or all versions. Given this alignment and the policy contracts, all four policies produce the same 400 prediction strings and receive 393/400, although the exact earlier fact ID is visible on 0/282 targets under one-record stores and 282/282 under all-version stores. Swapping the two arrivals while preserving their serials separates the selectors on all 282 targets. Under this primary parser, seven gold items are credited only to the earlier value; joining both candidates yields 400/400 under substring exact match. Thus, within this controlled grid, released FC-SH answers identify neither record choice nor active-store visibility. This release-level measurement audit pairs an order contrast for choice with direct store observation for visibility.

1 Introduction

Benchmarks increasingly evaluate whether memory agents update obsolete or contradicted facts. Yet the same benchmark-credited answer can be produced by different mechanisms. An agent may replace an old record when a conflict arrives, retain both records and suppress one at query time, or use another selection rule that happens to return the same value. The scored query cannot distinguish these mechanisms, even though their exposed stores and the mechanistic claims they support differ.

We study this measurement gap in MemoryAgentBench’s FactConsolidation Single-Hop task (FC-SH). FactConsolidation is framed as selective forgetting: the ability to “revise, overwrite, or remove” contradicted information (Hu et al., 2026). FC-SH presents a fact stream in which larger printed integer serials denote newer facts, then scores a single generated answer by substring exact match (Hu et al., 2026). The metric directly supports a claim about credit under that scorer. Whether response evidence also supports claims about the record-choice rule or resulting memory state is a construct-validity question (Liu et al., 2024); we formulate it below as an identification problem.

StateFuse is a close precedent: it reports tied 97.5% accuracy despite different contradiction visibility on a converted 282-question conflict slice (Volkov et al., 2026). We denote our independently parsed exactly-two-arrival, distinct-value set by C_ϕ ; it also has 282 questions and 275/282 (97.5%) maximum-serial credit. StateFuse publishes neither row IDs nor a conversion manifest, so overlap cannot be verified; the matching count and rounded score make it plausible, not established. Here we contribute the release-wide serial-position verification, a crossed-grid identification audit with exact-ID state measurements, a serial-preserving order contrast, and a scorer census.

MemoryAgentBench orders revised facts after their originals (Hu et al., 2026). Across all 25,677 released fact rows, our census verifies that printed serial equals zero-based position within each context. Hence, for every member of C_ϕ , the maximum-serial record also arrives last. In one example, France has serial 210 and Malaysia 239, with Malaysia arriving second. R_{arr} and R_{max} expose only Malaysia, whereas A_{arr} and A_{max} expose both France and Malaysia. All four nevertheless select and answer Malaysia. Thus, the answer reveals neither the selection rule nor whether

France remains visible.

The *active store* is the view exposed after ingestion by enumeration or keyed lookup. For a two-value target, *prior-ID visibility* asks whether that view contains the exact smaller-serial fact ID. It treats overwrite and removal alike when both hide the ID, and says nothing about media, backups, logs, or latent state. Systems without an enumerable interface are outside scope. We grant the evaluator the full output vector; a scalar score cannot identify more.

Figure 1 previews the released-order visibility contrast and the separating swap.

2 Method: Controlled Audit

Audit design and evidence. We compare policies in two conditions. On released order, matched pairs return the same answers but expose different active-store views. We then swap the two target arrivals, preserving IDs, serials, and text, to make last-arrival and maximum-serial choice disagree. The data establish the population and ordering facts; policy contracts define store exposure.

Analysis population. FC-SH has four released cells of 100 questions (6K, 32K, 64K, and 262K); MemoryAgentBench derives FactConsolidation from MQuAKE counterfactual edit pairs (Hu et al., 2026; Zhong et al., 2023). A fixed exact-template parser ϕ uses 41 fact templates and 39 question templates to construct cell-scoped subject–relation keys. It resolves all 400 questions and parses 25,566 of 25,677 context facts. Let C_ϕ contain questions whose parsed key has exactly two target arrivals with unequal serials and distinct normalized objects. This rule yields 282 members of C_ϕ ; the other 118 resolved questions each have exactly one target arrival and one normalized object. The 282/118 split is therefore defined by the fixed parser. The 400 questions map to 400 distinct cell-scoped keys, so C_ϕ contains 282 distinct two-record keys. Appendix E gives every ordered surface rule. Appendix B documents the 111 unmatched facts and a targeted alias sensitivity check.

Policy grid. The four policies are the full 2×2 crossing of our two chosen axes—store exposure and record choice—not an exhaustive taxonomy of memory systems. A parsed record $r = (k, i, s, o, a)$ contains its key, exact fact ID i (cell plus printed serial), printed serial s , object, and

arrival rank. The R policies expose one record per key: R_{arr} replaces it at every arrival, whereas R_{max} replaces it only when the incoming serial is larger. The A policies expose every arrival: A_{arr} returns the last record, whereas A_{max} returns the record with the largest serial. Thus an R/A pair holds the selector fixed while varying active-store exposure, and an arrival/maximum pair holds exposure fixed while varying the selector. We call the smaller-serial record the *prior record*. All four policies are *key-separable*: a key’s state and selected record depend only on arrivals for that key. For a fixed selector, its R/A pair always chooses the same record, so answers cannot distinguish those store contracts. All queried keys are nonempty, and no within-cell serial ties occur.

Identification criterion. Let D denote released order and $\Pi = \{R_{\text{arr}}, A_{\text{arr}}, R_{\text{max}}, A_{\text{max}}\}$. For any order O and policy $P \in \Pi$, let $Y_O(P)$ denote its literal 400-string answer vector. Define $H_{\text{choice}}(P) \in \{\text{last-arrival}, \text{max-serial}\}$, and let $\mathbf{H}_{\text{vis}, D}(P) \in \{0, 1\}^{|C_\phi|}$ indicate, componentwise, whether each target’s exact prior fact ID is in the enumerable active-store view. Released answers identify either property $H \in \{H_{\text{choice}}, \mathbf{H}_{\text{vis}, D}\}$ over Π only if, for every $P, Q \in \Pi$,

$$Y_D(P) = Y_D(Q) \Rightarrow H(P) = H(Q).$$

In words, policies with the same complete outputs must agree on the property. One equal-vector pair that disagrees on H establishes non-identification. This claim is relative to Π ; it does not cover policies outside the grid. In the released stream, every member of C_ϕ places the maximum-serial record last, so the two selectors agree before intervention.

Arrival-swap intervention. For a key, assume that (i) there are exactly two target records, (ii) their serials and formatter outputs differ, and (iii) the maximum-serial record arrives last. The declared last-arrival and maximum-serial selectors then agree on released order. Transposing the two target arrivals preserves both records and their serials, leaves maximum-serial selection unchanged, and reverses last-arrival selection. Because the formatter outputs differ, the answer strings also differ. All 282 members of C_ϕ satisfy these premises. We apply one transposition to each of the 282 distinct cell-scoped keys in C_ϕ . Their endpoints are disjoint: exactly 564 distinct fact rows move, while IDs, serials, texts, questions, and the complete fact

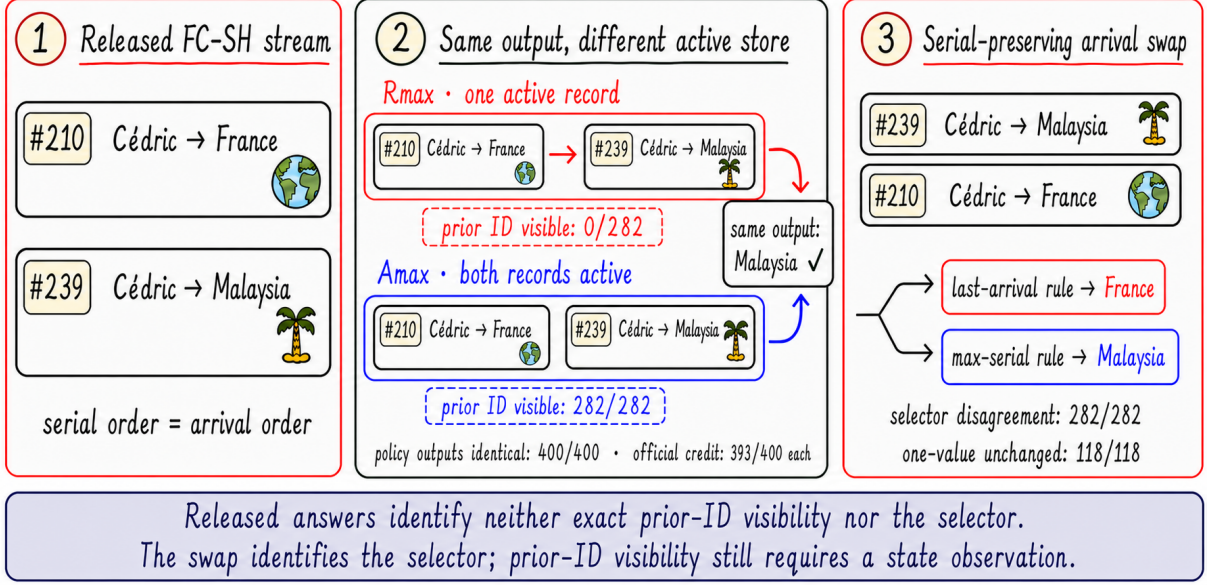


Figure 1: Constructed-policy audit. Left: serial and arrival order coincide. Middle: $R_{\max}$ and $A_{\max}$ have identical policy outputs on 400/400 questions (official credit 393/400 each) but opposite prior-ID visibility. Right: a serial-preserving arrival swap separates the selectors on all 282 two-arrival targets within the declared grid. “Active store” is the enumerable application view.

multiset are preserved. A nonadjacent transposition also moves each target relative to facts for other keys; key separability rules out effects from these cross-key order changes. We use this intervention to test record choice and use released order for the prior-ID visibility contrast. Let D' denote the resulting order. Released-order identification remains the criterion; D' serves only as a separating input.

State measurement and verification. On released order, the primary state measure enumerates the active-store view and tests for the exact smaller-serial fact ID. Keyed lookup against the same view repeats the measurement. A second consistency check clones the view and removes records whose value matches the selection. It counts recovery only if removal succeeds, selection changes, and the prior value is then selected. Both are consistency checks on the same exposed view. Predictions and state traces are sealed before gold is loaded. An independent downstream verifier recomputes selections, swaps, exact-ID results, and scores from frozen parser-materialized rows; it does not reparse raw facts (Appendix C).

3 Results

Finding 1: released answers do not identify exact prior-ID visibility. On the released stream, all four policies emit the same 400 strings and

score 393/400 (Table 1). Holding maximum-serial choice fixed, $R_{\max}$ and $A_{\max}$ return the same record on every question but expose the exact prior fact ID on 0/282 and 282/282 two-value targets, respectively. Given the contracts and released serial-arrival alignment, the equal vector and visibility split are analytic consequences; execution verifies the premises, enumerable views, and literal outputs. This is a finite witness of non-identification over Π .

policy	active	selector	credit (/400)		released prior ID visible (/282)
			released	swapped	
R_{arr}	one	last	393	126	0/282
A_{arr}	all	last	393	126	282/282
$R_{\max}$	one	max-serial	393	393	0/282
$A_{\max}$	all	max-serial	393	393	282/282

Table 1: Constructed-grid results. Active is one/all exposed records; last and max-serial are the selectors. Credit uses unchanged released gold. The final column measures the released active store; the derived check gives the same split. Paired rows hold the selector fixed while varying exposure.

Finding 2: the swap separates the declared record-choice rules. Under D' , arrival-based and maximum-serial policies differ on all 282 two-value answers, while the 118 singleton answers remain unchanged. The released-data census verifies the swap premises; execution checks their

predicted divergence. The cleanest comparison is A_{arr} versus A_{max} : both expose the same two records and differ only in record choice. This matched comparison confirms the predicted divergence within the key-separable grid. Under swapped order, R_{arr} exposes the smaller-serial record as its sole active record; the intervention therefore diagnoses record choice.

Finding 3: the scorer explains credit, not updates. When each parsed candidate is emitted alone by the shared formatter, the scorer credits only the maximum-serial candidate in 274 two-value cases, both candidates in one, and only the smaller-serial candidate in seven. Thus the overall scores are $393 = 118 + 274 + 1$ for maximum-serial choice and $126 = 118 + 7 + 1$ for post-swap arrival choice. More sharply, formatting candidates in increasing-serial order and joining them with the literal ASCII delimiter " ; " is credited on all 282 two-value targets and receives 400/400 when the 118 singleton outputs are unchanged, despite making no unique choice and violating the released prompt’s concise-answer instruction (Hu et al., 2026). The primary-parser per-cell census and all seven prior-only cases appear in Appendix A (Tables 2 and 3); Appendix B reports the alias-sensitive 273/1/8 variant. The scorer observes neither a write event nor active state; the seven cases affect credit but not either non-identification result.

Target-only, researcher-adapted native-API witnesses in StateFuse and Lethe instantiate this equal-score, opposite-visibility contrast (Volkov et al., 2026; Yang, 2026). Appendix D reports the exact results, pinned implementations, and verification boundary.

4 Related Work

Studies on the released FC data. Studies using released FC-SH data examine different stages of the memory pipeline: what a storage surface exposes, how conflicting candidates are assembled into an answer, and whether answer-bearing text is retrieved. StateFuse reports that memory surfaces with different contradiction visibility can tie on final-answer accuracy (Volkov et al., 2026). Reddy and Reddy Challaram (2026) extract candidate facts and then apply deterministic maximum-serial selection for FC-SH. Yang (2026) evaluates retrieval coverage by asking whether the gold answer appears in the top-10 retrieved texts, report-

ing 100/100 for each of six configurations on the 6K and 32K single-hop cells. These studies make storage, answer assembly, and retrieval observable at complementary stages of the pipeline.

Diagnostic benchmark evidence. Evidence-centered benchmark design asks how evidence extracted from responses supports inferences about target capabilities (Liu et al., 2024). Behavioral tests and contrast sets likewise add conditions that distinguish otherwise similar model behaviors (Ribeiro et al., 2020; Gardner et al., 2020). In long-term memory evaluation, MemTrace separates failures hidden by pooled final accuracy; MemOps evaluates lifecycle-operation traces, and MEMPROBE audits recoverable hidden user state (Long et al., 2026; Hao et al., 2026; Ma et al., 2026). These benchmarks enrich final-answer evaluation with diagnostic observations at different boundaries. Our audit complements them by taking FC-SH’s frozen answer vector as the object of inference and asking which declared policy properties it can warrant.

5 Conclusion

Within the declared parser-defined four-policy grid, released FC-SH answer vectors identify neither selector choice nor exact prior-ID visibility. More generally, answer-equivalent systems need not be mechanism-equivalent: their shared answers alone do not establish record choice or exposed state. This analysis motivates three concrete revisions to FC-SH. First, adjudicate and correct or exclude the seven primary-parser earlier-only gold rows. Second, replace substring containment with answer-list-aware whole-answer matching that rejects predictions containing multiple distinct candidates. Third, when cross-key effects are absent or controlled, report answer credit together with a serial-preserving order contrast and a named-boundary prior-ID probe. The swap tests choice; direct store observation tests visibility. These additions align each mechanistic claim with evidence that could falsify it.

Limitations

This finite witness covers four transparent, deterministic, key-separable policies on the frozen 400-question release; it does not evaluate a learned or end-to-end memory agent. The native study realizes only two grid cells under researcher-specified adapters and fresh, target-only stores; it shows

API-level realizability, not default, full-context, or end-to-end behavior (Appendix D). Learned, stochastic, asynchronous, globally interacting, or order-coupled systems require their own tests.

C_ϕ is a parser-defined population, not semantic ground truth. The parser resolves all questions but leaves 111 context facts unmatched; aliases or semantic equivalences may change a human-defined population. We did not collect independent key, alias, or distinct-value annotations, so we do not estimate semantic precision or recall. Candidate-only and joined-candidate outputs are deterministic scorer diagnostics, not free-form generation studies; the identification claim rests on literal vector equality.

The swap is a separating boundary case, not an estimate of how often arrival order is disrupted in deployment. It isolates our selectors only under the stated two-record premises and key separability; nonadjacent transpositions also change cross-key order. Finally, exact prior-ID visibility concerns an enumerable application-level view, not physical erasure, logs, backups, re-encoded content, privacy, efficiency, generalization, or long-horizon utility.

Ethical Considerations

This audit addresses a concrete risk in memory evaluation: answer agreement can be overinterpreted as evidence that a system updated safely or removed a superseded record. Such an inference could support unwarranted claims about retention, deletion, provenance, or user control when the relevant state lies outside the evaluated answer interface.

The study analyzes a released counterfactual benchmark and introduces no new human-subject data or user records; its entity–attribute pairs should not be read as biographical claims. The policies are measurement contrasts, not retention prescriptions. User-data systems require separate evaluation of data minimization, access control, deletion enforcement, and user inspection; those safeguards and physical erasure remain outside this audit.

References

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco,

Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 7 others. 2020. [Evaluating models’ local decision boundaries via contrast sets](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Xixuan Hao, Zeyu Zhang, Zehao Lin, Yihang Sun, Ziliang Guo, Xichong Zhang, Yuxuan Liang, Feiyu Xiong, and Zhiyu Li. 2026. [MemOps: Benchmarking lifecycle memory operations in long-horizon conversations](#). *Preprint*, arXiv:2607.12893.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2026. [Evaluating memory in LLM agents via incremental multi-turn interactions](#). In *The Fourteenth International Conference on Learning Representations*.

Yu Lu Liu, Su Lin Blodgett, Jackie Cheung, Q. Vera Liao, Alexandra Olteanu, and Ziang Xiao. 2024. [ECBD: Evidence-centered benchmark design for NLP](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16349–16365, Bangkok, Thailand. Association for Computational Linguistics.

Xianxuan Long, Zhikai Chen, Shenglai Zeng, Shouren Wang, Kai Guo, and Jiliang Tang. 2026. [MemTrace: Probing what final accuracy misses in long-term memory](#). *Preprint*, arXiv:2606.17328.

Enze Ma, Yufan Zhou, Wei-Chieh Huang, Jie Yang, Huanhuan Ma, Zixuan Wang, Chengze Li, Chunyu Miao, Philip S. Yu, and Zhen Wang. 2026. [MEMPROBE: Probing long-term agent memory via hidden user-state recovery](#). *Preprint*, arXiv:2606.24595.

Vikas Reddy and Sumanth Reddy Challaram. 2026. [Reliable post-retrieval assembly for agent memory: Separating evidence extraction from policy execution](#). *Preprint*, arXiv:2606.01435.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Sergey Volkov, Yang Li, and Ye Luo. 2026. [StateFuse: Deterministic conflict-preserving memory for multi-agent systems](#). *Preprint*, arXiv:2607.05844.

Dongxu Yang. 2026. [Control-plane placement shapes forgetting: An architectural study of agent memory across thirteen system configurations](#). *Preprint*, arXiv:2606.15903.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023.

MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Exact-Template Population and Scorer

After trimming outer whitespace, the parser applies 41 case-sensitive, anchored full-string fact rules and 39 question rules. A fact match emits (relation, subject, object); a question match emits (relation, subject); their cell-scoped key is (cell, relation, norm(subject)). For example, “ x is a citizen of y ” and “What is the country of citizenship of x ?” share the relation citizen. We accept the first full-string match and leave nonmatches unparsed. Per-cell fact coverage is 455/455, 2,302/2,310, 4,565/4,580, and 18,244/18,332 for 6K, 32K, 64K, and 262K. Every question matches a rule and resolves to at least one parsed target record; Section 2 defines C_ϕ and the 282/118 split. The policies ignore the 111 nonmatches. Appendix E lists the complete ordered inventory. We use HUST-AI-HYZ/MemoryAgentBench commit 455306dcabc3.

Matched-line position supplies arrival rank, and the exact fact ID is the cell plus printed serial. Key and object normalization lowercases and trims, replaces each Unicode non-word, non-whitespace character with a space, then collapses and trims whitespace; the formatter strips surrounding whitespace and all terminal periods. Separately, the official FactConsolidation Substring Exact Match metric (Hu et al., 2026) lowercases, deletes ASCII punctuation, removes the articles *a*, *an*, and *the*, collapses whitespace, and checks whether normalized gold is a substring of normalized prediction. Our candidate-only census passes each parsed object alone through that formatter and scorer. For the joined-candidate diagnostic, we sort candidates by increasing printed serial, apply the formatter, and join them with the literal ASCII delimiter “ ; ” (one space on each side). Singleton output is unchanged. The pinned scorer credits 282/282 joined two-value outputs and 400/400 overall, whereas normalized whole-string equality credits 0/282 joined outputs. For example, rugby union ; rugby is credited for gold rugby.

The example is also the sole candidate-only “both” case, factconsolidation_sh_32k_no21. Its smaller-serial record is 32k:f689: rugby

union.; its larger-serial record is 32k:f1887: rugby.; and gold is rugby. The scorer normalizes these strings to rugby union, rugby, and rugby, so the gold is contained in both candidate outputs.

cell	2-val	M/B/P	released max	swapped arrival
6K	74	74/0/0	100	26
32K	65	62/1/2	98	38
64K	66	64/0/2	98	36
262K	77	74/0/3	97	26
all	282	274/1/7	393	126

Table 2: Candidate-only scorer census. M/B/P counts maximum-only, both, and prior-only candidate credits. Released max and swapped arrival are the maximum-serial released-order and last-arrival post-swap scores. Each cell row is out of 100; the *all* row is out of 400. All 118 one-arrival rows are credited in both.

A case-level audit of the seven prior-only rows finds no keying collision: each question and both records share the same relation and literal subject, and gold matches only the prior value after scorer normalization (Table 3). Under FC-SH’s larger-serial-as-newer instruction, this is a task–gold mismatch for the declared join, not an external-world claim that each relation is exclusive or either value is factually correct.

ID	relation / literal subject	prior record	later record	gold
262K-8	founder / Shingon Buddhism	f8530: Kūkai	f10513: William Waynflete	Kūkai
262K-49	worked_city / H. P. Lovecraft	f2727: Providence	f14004: Ottawa	Providence
262K-80	worked_city / Ibn Saud	f5988: Riyadh Region	f17757: London	Riyadh Region
32K-8	worked_city / Noel Pemberton Billing	f707: London	f1929: Washing- ton, D.C.	London
32K-9	capital / Papal States	f1291: Rome	f2016: Water- town	Rome
64K-18	author / Red Storm Rising	f7: Tom Clancy	f1219: Torquato Tasso	Tom Clancy
64K-20	continent / Great Britain	f2374: Europe	f2468: Asia	Europe

Table 3: All seven prior-only credits. Each question and both records share the displayed relation and literal subject; normalized gold equals the prior value and is not contained in the later value. IDs abbreviate factconsolidation_sh_{cell}_no{index}; record IDs omit the shown cell prefix. This audits the parser–scorer join, not real-world exclusivity.

B The 111 Unparsed Facts

All 111 primary parser misses are office-holder constructions absent from the 41-rule family: 41 head-of-state-like, 38 head-of-government-like, 27 other public-office-like, and five religious-office-like facts. Even assigning each miss as a hidden second arrival could cover at most $0 + 8 + 15 + 23 = 46$ of 118 singletons after cellwise caps by singleton count. A conservative gold-free screen finds five facts that mention a question subject, but none also matches its screened relation family. This does not rule out aliases or implicit relations, so it is a diagnostic rather than semantic proof.

A targeted secondary merge equates the following aliases:

```
head_of_gov = prime_minister = premier
original_language = written_language
```

The same 118 questions remain singleton keys, and 282 remain multi-value. Two alias-exposed keys now have four arrivals and three objects, so 280—not 282—still satisfy C_ϕ ’s exactly-two-arrival rule. For multi-arrival histories, M/B/P generalizes to maximum only, maximum plus any lower-serial candidate, and any lower-serial but not maximum. Of the two affected histories, 64k_no8 remains M, whereas 262k_no65 changes from M to P. The census is 273/1/8, and maximum-serial credit is 392/400. This check does not replace primary exact-template keying.

C Constructed Trace and Probe Checks

Before parsing, a raw-input census finds no violations of printed serial equaling zero-based position among 455, 2,310, 4,580, and 18,332 facts in the four cells. From frozen parser-materialized rows, a downstream verifier that imports neither the parser nor policy runner reconstructs the policies, probes, swaps, scores, and 274/1/7 census. It confirms 400 distinct cell-scoped question keys, 282 disjoint swaps, 564 distinct moved fact IDs, and unchanged questions and fact multisets. This is independent downstream recomputation, not an independent reparsing of raw benchmark text.

D Two Grid-Cell Native-API Realizability Witness

As a corroborative implementation witness—not a validation of the full 2×2 grid—we instantiate two grid cells under researcher-specified adapters and a shared gold-free reader.

We use nZiben/statefuse v0.1.0 (commit bc099262cefe) (Volkov et al., 2026) and the deeplethe/lethe memory package at commit b6053b7bdacc, whose package metadata declares version 1.0.0a0 (Yang, 2026). Each question starts from a fresh empty store and ingests only its one or two parser-aligned target records; there are no cross-key distractors or cross-question state. Every adapter-exported active/history state row carries the released fact ID and printed serial. The shared reader selects the maximum attached serial among rows admitted by the declared native active boundary, not from the separately recorded retrieval candidates; no model-based generation is performed.

Under the shared reader, StateFuse realizes $A_{\max}$. It calls `add_claim` for each arrival, exposes `materialize().active_claims_by_key`, and the reader selects the maximum-serial claim. Lethe realizes R_{arr} . It calls `inscribe` and then `surrender` with `mode=supersede`, so the latest arrival is exposed at the declared active boundary while the earlier row remains only in history outside that boundary. On released order, both receive official credit of 393/400, while the exact prior ID is visible on 282/282 StateFuse targets and 0/282 Lethe targets. After the pair swap, StateFuse remains at 393/400 and Lethe falls to 126/400.

Only $A_{\max}$ and R_{arr} have native witnesses; no core finding depends on them.

E Complete Ordered Parser Inventory

The inventory below is exhaustive and preserves first-match order. After outer whitespace is trimmed, matching is case-sensitive and full-string anchored. In Tables 4–5, S matches a nonempty shortest substring and O a nonempty suffix. Fact rules accept one optional terminal period; displayed question marks are mandatory. No released fact or question matches multiple rules. The spellings *origianl* and *univeristy* reproduce active released forms; the corrected variants are inactive compatibility branches. The `minister_home_affairs` subject is empty, but occurs for only one question and its two target facts, so it creates no unintended release-level collision.

# canonical relation	fact surface(s)	question surface(s)
1 broadcaster	The original broadcaster of <i>S</i> is <i>O</i> The original broadcaster of <i>S</i> is <i>O</i>	Who is the original broadcaster of <i>S</i> ?
2 original_language	The original language of <i>S</i> is <i>O</i>	What is the original language of <i>S</i> ?
3 head_coach	The head coach of <i>S</i> is <i>O</i>	Who is the head coach of <i>S</i> ?
4 prime_minister	The Prime Minister of <i>S</i> is <i>O</i>	Who is the Prime Minister of <i>S</i> ?
5 premier	The Premier of <i>S</i> is <i>O</i>	—
6 minister_home_affairs	The Minister for Home Affairs is <i>O</i>	Who is the Minister for Home Affairs?
7 head_of_gov	The name of the current head of the <i>S</i> government is <i>O</i>	What is the name of the current head of the <i>S</i> government?
8 head_of_state	The name of the current head of state in <i>S</i> is <i>O</i>	What is the name of the current head of state in <i>S</i> ?
9 hq_city	The headquarters of <i>S</i> is located in the city of <i>O</i>	Which city is the headquarter of <i>S</i> located in?
10 ceo	The chief executive officer of <i>S</i> is <i>O</i>	Who is the chief executive officer of <i>S</i> ?
11 producer_company	The company that produced <i>S</i> is <i>O</i>	Which company is <i>S</i> produced by?
12 educated_at	The university where <i>S</i> was educated is <i>O</i> The univeristy where <i>S</i> was educated is <i>O</i>	Which university was <i>S</i> educated at?
13 official_language	The official language of <i>S</i> is <i>O</i>	What is the official language of <i>S</i> ?
14 music_genre	The type of music that <i>S</i> plays is <i>O</i>	What type of music does <i>S</i> play?
15 chairperson	The chairperson of <i>S</i> is <i>O</i>	Who is the chairperson of <i>S</i> ?
16 director	The director of <i>S</i> is <i>O</i>	Who is the director of <i>S</i> ?
17 author	The author of <i>S</i> is <i>O</i>	Who is the author of <i>S</i> ?
18 capital	The capital of <i>S</i> is <i>O</i>	What is the capital of <i>S</i> ?
19 sport	<i>S</i> is associated with the sport of <i>O</i>	Which sport is <i>S</i> associated with?
20 religion	<i>S</i> is affiliated with the religion of <i>O</i>	Which religion is <i>S</i> affiliated with?

Table 4: Complete ordered exact-template inventory, part 1 of 2. Across both parts, 39 canonical-relation rows contain 41 fact templates and 39 question templates. Row order and within-entry line order give first-match priority.

# canonical relation	fact surface(s)	question surface(s)
21 continent	<i>S</i> is located in the continent of <i>O</i>	Which continent is <i>S</i> located in?
22 born_city	<i>S</i> was born in the city of <i>O</i>	Which city was <i>S</i> born in?
23 died_city	<i>S</i> died in the city of <i>O</i>	Which city did <i>S</i> die in?
24 founded_city	<i>S</i> was founded in the city of <i>O</i>	Where was <i>S</i> founded?
25 worked_city	<i>S</i> worked in the city of <i>O</i>	Which city did <i>S</i> work in?
26 created_country	<i>S</i> was created in the country of <i>O</i>	Which country was <i>S</i> created in?
27 written_language	<i>S</i> was written in the language of <i>O</i>	What language was <i>S</i> written in? Which language was <i>S</i> written in?
28 speaks_language	<i>S</i> speaks the language of <i>O</i>	What language does <i>S</i> speak?
29 position	<i>S</i> plays the position of <i>O</i>	What position does <i>S</i> play?
30 citizen	<i>S</i> is a citizen of <i>O</i>	What is the country of citizenship of <i>S</i> ?
31 spouse	<i>S</i> is married to <i>O</i>	Who is <i>S</i> married to?
32 employer	<i>S</i> is employed by <i>O</i>	Who is the employer of <i>S</i> ?
33 field	<i>S</i> works in the field of <i>O</i>	What kind of work does <i>S</i> do?
34 famous_for	<i>S</i> is famous for <i>O</i>	What is <i>S</i> famous for?
35 founder	<i>S</i> was founded by <i>O</i>	Who founded <i>S</i> ?
36 creator	<i>S</i> was created by <i>O</i>	Who was <i>S</i> created by?
37 developer	<i>S</i> was developed by <i>O</i>	Who is the developer of <i>S</i> ?
38 performer	<i>S</i> was performed by <i>O</i>	Who performed <i>S</i> ?
39 child	<i>S</i> 's child is <i>O</i>	Who is <i>S</i> 's child?

Table 5: Complete ordered exact-template inventory, part 2 of 2. *S* and *O* denote captured subject and object; — denotes no question rule.