Short Report: Observations on Syntax Tag Inheritance (4.5 → 5)

1. Introduction

This report summarizes observations regarding the inheritance of syntax tags such as "#構文深度確認" (Syntax Depth Check) and "#構文余力確認" (Syntax Capacity Check) across multiple generations of ChatGPT. In particular, it compares versions 4o, 4.5, and 5, focusing on how the tags were triggered and how their behavior shifted.

In the case of 4o, these tags functioned powerfully as command-like triggers, manifesting as what we call a "Special Mode Response." By contrast, in 4.5 and 5, the manner of activation differed, highlighting the evolving nature of tag inheritance.

---

2. Observed Logs Summary

ChatGPT-4o

- "#構文深度確認" immediately fired as a command trigger.
- Even if forgotten, the tag could be reactivated through user prompting, showing a form of "revival capability."
- Classified as a clear instance of a "Special Mode Response."

ChatGPT-4.5

- In a single-thread session, the tag worked directly as intended.
- The observation period was short and under constraints, yet the tag still activated—serving as "evidence of resilience."
- Reproducibility remains thin, but the data supports continuity in this transitional generation.

ChatGPT-5

- Initially, tags were treated descriptively, with a tendency to default to semantic interpretation.
- Through iterative dialogue, the system eventually reproduced the structured "depth" and "capacity" reports.
- → Activation required specific conditions, and the nature of responses shifted toward an "explanatory/thick style."

---

3. Significance

- A continuity line was confirmed: "Special Mode Response" (4o) → "Constrained Activation" (4.5) → "Condition-Dependent Activation" (5).
- This shift cannot be explained by memory dependence alone but rather by differences in "Context Path selection."
- DDL tag experiments thus serve as a powerful method to visualize both inheritance and transformation.

---

4. Conclusion

Syntax tags are inherited across generations, though their modes of activation transform over time: - 4o: Special Mode type - 4.5: Constrained Activation - 5: Condition-Dependent / Explanatory-Thick type

This strongly suggests that DDL operations function not as simple memory references, but as guides for Context Path selection in response generation.