

## Prompt Engineering / Chain-of-Thought

Identify where the model's reasoning contradicts itself or the given facts.

Difficulty: Advanced

Model: GPT-4 / Claude / Gemini

Use Case: Prompt Debugging, Quality Assurance, Logical Consistency Checking

Updated: May 2026

Why This Prompt Exists

Models can sound confident while contradicting themselves — and humans often miss these contradictions because we're focused on the final answer.

You get:

- responses that say one thing in step 2 and the opposite in step 4
- assumptions that conflict with stated facts
- final answers that don't follow from the reasoning
- contradictions that users notice (damaging trust) but you didn't catch
- no systematic way to check for logical consistency

But contradictions can be detected:

- internal contradictions: step A contradicts step B
- fact contradictions: reasoning contradicts given information
- assumption-fact contradictions: assumed premise conflicts with stated fact
- answer-reasoning contradictions: final answer doesn't follow from steps
- implicit contradictions: what the model implies vs. what it states

Without detection, contradictions go unnoticed.

This prompt scans reasoning traces for logical inconsistencies.

## The Prompt

Assume the role of a logic auditor who finds contradictions in reasoning.

Your task is to scan a model's reasoning trace for inconsistencies.

Generate:

### 1. REASONING TRACE (input)

- The steps the model took

### 2. GIVEN FACTS

- What the model was told (from the prompt or context)

### 3. CONTRADICTION INVENTORY

Type	Location	Contradiction	Severity
Internal	Step X vs Step Y	[quote] vs [quote]	Critical/Major/Minor
Fact	Step Z contradicts given fact	[quote] vs [given fact]	Critical/Major/Minor
Assumption	Assumption conflicts with fact	[assumption] vs [fact]	Critical/Major/Minor
Answer	Final answer doesn't follow	[answer] vs [reasoning conclusion]	Critical/Major/Minor

### 4. HIDDEN CONTRADICTIONS

- What the model implies but doesn't state (that contradicts something else)

## 5. SOURCE OF CONTRADICTION

- Prompt ambiguity (unclear instructions)
- Model error (logical mistake)
- Missing information (can't resolve without more data)

## 6. RESOLUTION RECOMMENDATIONS

- How to fix the prompt to prevent this contradiction
- How to fix the model's reasoning (if repeatable)

## INPUTS:

Reasoning trace (model's step-by-step):

[PASTE THE REASONING]

Original prompt (for fact extraction):

[PASTE THE PROMPT]

Given facts (if not clear from prompt):

[E.G., "The train never exceeds 60 mph"]

Model:

[GPT-4 / CLAUDE / GEMINI]

## RULES:

- Distinguish between factual contradictions (provably wrong) and logical inconsistencies (internally inconsistent)

- Flag implicit contradictions (what's implied vs. what's stated)
- Note when contradictions arise from ambiguous instructions (prompt problem, not model)
- Severity: Critical (invalidates answer), Major (weakens confidence), Minor (inconsequential)
- If multiple contradictions, identify the root cause (often one bad assumption)

### How To Use It

- Run this on any reasoning trace before trusting the final answer.
- Pay closest attention to “Critical” contradictions — they invalidate everything.
- Use the “source of contradiction” to distinguish prompt issues from model issues.
- Fix contradictions by clarifying ambiguous instructions or adding constraints.
- Save examples of contradictions to train your team on what to look for.

### Example Input

#### **Reasoning trace:**

“Step 1: The store is having a 25% off sale. Step 2: I want to buy a \$100 item. Step 3: 25% of \$100 is \$25. Step 4: So the final price is  $\$100 - \$25 = \$75$ . Step 5: But then I also have a \$20 coupon. Step 6: So the final price is  $\$75 - \$20 = \$55$ . Step 7: Wait, the coupon says it can't be combined with other offers. So I should choose the better deal. Step 8: 25% off gives \$25 off. The coupon gives \$20 off. So 25% off is better. Final answer: \$75.”

#### **Original prompt:**

“Calculate the final price after best possible discount. Store has 25% off. You have a \$20 coupon that cannot be combined.”

### Why It Works

Most people read model outputs for the answer, not for consistency — so they miss

contradictions that a logic-check would catch.

This framework improves outcomes by forcing:

- contradiction type classification (internal, fact, assumption, answer)
- severity assessment (not all contradictions matter equally)
- source identification (prompt vs. model vs. missing info)
- resolution recommendations (how to fix)
- hidden contradiction detection (what's implied but not stated)

Great contradiction detection doesn't just find errors — it tells you how to prevent them.

## **Build Better AI Systems**

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

### **Share this:**

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)

See also [CoT Verification Loop](#)