Lógica na Era dos LLMs: Por que o ‘Logo’ Ainda é o Alicerce de Tudo

Hoje, Já estamos vivendo imersos em Modelos de Linguagem de Grande Escala (LLMs) que parecem ”entender” o mundo de forma quase Mágica.mas essa ”mágica” não é baseada em determinismo logico, e sim em probabilidades. O que

2025-11-10

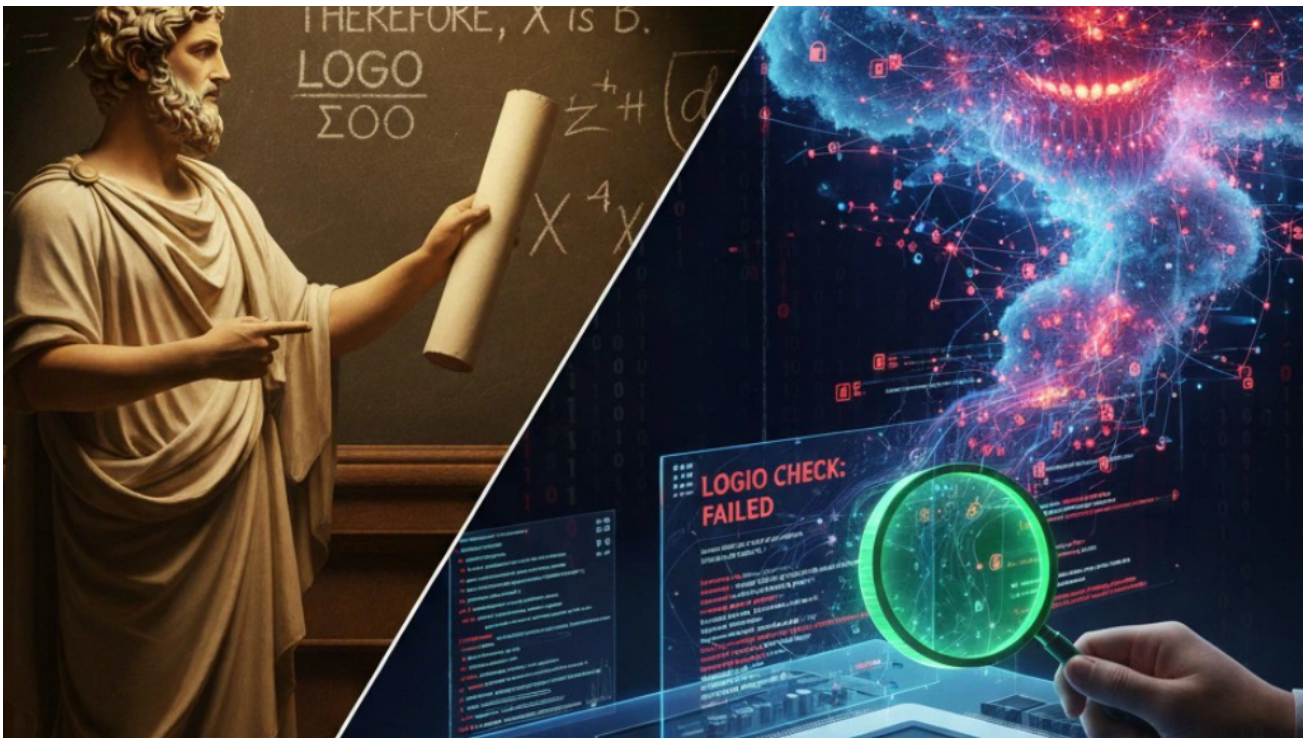


Figure 1: Capa do artigo

Hoje, Já estamos vivendo imersos em Modelos de Linguagem de Grande Escala (LLMs) que parecem “entender” o mundo de forma quase Mágica.mas essa “mágica” não é baseada em determinismo logico, e sim em probabilidades. O que nos leva ao lugar que a lógica não é apenas relevante; ela é o alicerce.

A lógica clássica, como é a ferramenta que usamos para construir garantias. A estrutura é a seguinte:

```
Todo A é B.
X é A.
Logo, X é B.
```

Se as premissas (A é B, X é A) são verdadeiras, a conclusão (X é B) é inegavelmente verdadeira.

Vamos pegar um exemplo mais do dia a dia de TI

```
"Permissão = (O usuário é 'Admin') OU (O usuário tem
```

© Artemisia Weyl 2024–2026 — Todos os direitos reservados.

```
'Assinatura Ativa' E a feature flag 'NovoDashboard' está
ligada)
```

Para o computador, isso vira uma equação.

Vamos dizer que:

- A = Usuário é Admin (Verdadeiro/Falso)
- B = Usuário tem Assinatura Ativa (Verdadeiro/Falso)
- C = Feature Flag 'NovoDashboard' está ligada (Verdadeiro/Falso)

A regra de negócio torna-se uma equação lógica, usando (E) e + (OU):

$$\text{Permissão} = A + (B \cdot C)$$

O computador, então, apenas calcula o resultado com base nas regras fundamentais: Verdadeiro + Falso = Verdadeiro (1 + 0 = 1) Verdadeiro. Falso = Falso (1 * 0 = 0). Verdadeiro Verdadeiro = Verdadeiro (1 * 1 = 1)

Este é o ponto crucial para 2025, e onde mora o perigo:

O seu LLM pode sugerir essa linha de código (Permissao = A + (B * C)). Mas, por ser uma ferramenta estatística, ele pode gerar um código que parece perfeito, mas contém uma falha lógica sutil.Ele pode, por exemplo, esquecer os parênteses (B * C), gerando A + B * C, o que pode falhar ou ser avaliado incorretamente dependendo da linguagem. Ou, muito pior, pode introduzir uma vulnerabilidade de segurança (como um SQL injection ou um XSS) por ter aprendido esse padrão em milhões de exemplos de código inseguro da internet.

É aqui que o nosso papel se torna crítico. A CPU, ao executar o código, e a API, ao proteger o dashboard, operam apenas com a lógica booleana determinística. Elas vão executar o código defeituoso ou inseguro que o LLM escreveu com a mesma eficiência.

Cabe a nós, os humanos, sermos os validadores finais dessa lógica.

Em 2025, o maior perigo profissional não é ser substituído por um LLM. O perigo é tornar-se um apertador de botão: Alguém que aceita a saída probabilística da IA e a trata como uma verdade lógica e determinística.

Como vimos, a IA gera o que é estatisticamente provável, não o que é logicamente garantido. Ela vai "alucinar" bugs sutis, vai introduzir vulnerabilidades de segurança que parecem código legítimo e vai criar falhas de lógica de negócio complexas.

Confiar cegamente nesse código é abdicar da nossa principal função. A lógica humana, estudada desde 200 AC, e que não deve ser substituída. Ela se torna nossa ferramenta de auditoria mais crítica.

Nós, como engenheiros, desenvolvedores e arquitetos, somos os guardiões dessa validação. O trabalho não é mais apenas escrever lógica, mas garantir que a lógica, venha ela de um cérebro humano ou de um modelo de silício, esteja correta.

Postagem original no LinkedIn: Lógica na Era dos LLMs: Por que o "Logo" Ainda é o Alicerce de Tudo

© Artemisia Weyl 2024–2026 — Todos os direitos reservados.