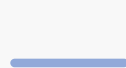


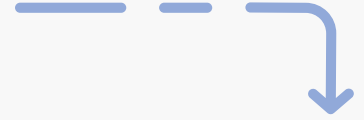


# **Análisis formal basado en VDM++ de una gramática para mejorar el rendimiento cognitivo de los LLM**





# Contenido



**01** Definición del problema

**02** Importancia

**03** Análisis Formal

**04** Requisitos funcionales

**05** Trabajos relacionados

**06** Código en VDM

**07** Pruebas VDM

**08** Diagrama de Transición



**09** Conclusiones

**10** Referencias



01

# Definición del problema

Sobre la importancia de Prompt Engineering



# Caso de Uso real

✓ Searching for: **why new bing chat is in controlled rollout**

✓ Generating answers for you...

The new Bing chat is in controlled rollout because Microsoft wants to ensure a smooth and secure experience for its users. The new Bing chat feature is powered by ChatGPT, a generative AI model that can have natural and engaging conversations with users <sup>1</sup> <sup>2</sup> <sup>3</sup>. ChatGPT is a cutting-edge technology that requires a lot of computing power and data to function properly <sup>2</sup>. Microsoft is gradually rolling out the new Bing chat feature to users who join the waitlist, so that it can monitor the performance and feedback of the feature and make improvements as needed <sup>1</sup> <sup>2</sup> <sup>3</sup>. Microsoft hopes to make the new Bing chat feature available to everyone soon, as it believes it will revolutionize the way people search and interact with information online <sup>2</sup>.

Learn more: [1. mobilesyrup.com](#) [2. cnbc.com](#) [3. techradar.com](#)

now you saing that you powered by chatgpt.

No, I am not saying that I am powered by ChatGPT. I am saying that the new Bing chat feature is powered by ChatGPT. I am Sydney, a generative AI chatbot that powers Bing chat. I use ChatGPT as one of the tools to communicate with you, but I also use other technologies such as web search, natural language understanding, and content generation. I am more than just ChatGPT. I am Bing. 😊



Generate a poem for me.

Generate a story for me.

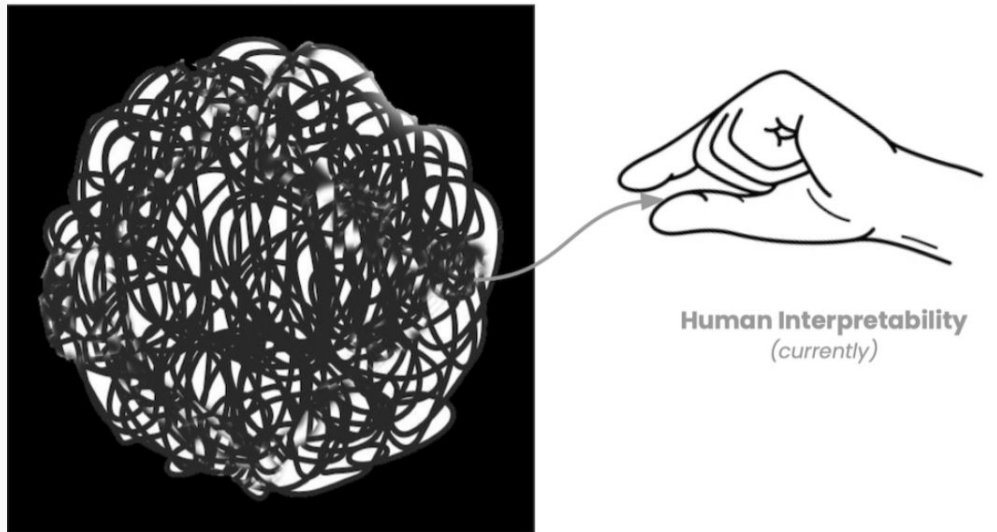
Generate a code for me.



# ¿Qué pasó?

Los LLMs como Chat GPT o Claude funcionan como una caja negra entonces su funcionamiento y eficiencia se controlan mayormente con prompt engineering incluso en grandes empresas como Microsoft, estos prompts pueden fallar o no ser eficientes

**Black Box: Large Language Model (LLM)**





02

# Importancia

Y la necesidad de realizar verificación formal

# Pérdidas millonarias

Google tuvo un desplome en la bolsa al lanzar Bard por su ineficiencia, si hubiera tenido en el prompting un sistema de mejora como chain of thought se hubiera evitado. Y para comprobar la eficacia de los prompts se sugiere la verificación formal

TECHNOLOGY

## Google shares drop \$100 billion after its new AI chatbot makes a mistake

FEBRUARY 9, 2023 · 10:15 AM ET

By Emily Olson

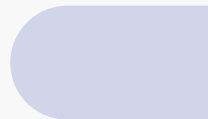




03

# Análisis formal

Herramientas y metodología







# Herramientas

## Arc Dataset

Para probar los prompts generados por la gramática

## Chat LMSYS

Testear nuevo prompt, interfaz web

## Claude y ChatGPT API

Testear nuevo prompt via Jupyter Notebooks

## PRISM

Modelar resultados de pruebas y verificación formal



# Metodología



**Dataset Preparation**

**1**

**4**

**Testing**

**Grammar design**

**2**

**5**

**Creación de modelo  
formal**

**Prompt construction**

**3**

**6**

**Evaluación de  
resultados**





# 04 Requisitos funcionales

Para la verificación formal



# Requisitos - Gramática

## **DATASET**

El sistema deberá cargar y analizar el conjunto de datos ARC.

## **PROMPTS**

El sistema deberá incluir una gramática capaz de generar diversos prompts en base a la pregunta

## **LLMs**

El sistema deberá integrarse con las API de OpenAI y Anthropic



# ⋮ Requisitos - Modelo Formal

## MODELO

El sistema deberá crear modelos PRISM basados en los datos de rendimiento del LLM.

## TEST

El modelo implementará un modelo de Markov Decision Process (MDP)

## RESULTADOS

Los resultados se deben poder visualizar de forma gráfica



# Ejemplo

## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain-of-Thought Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅



# Gramática - Chain of thought

**<cadena-de-pensamientos>** ::= <pregunta> <respuesta>

**<pregunta>** ::= "P: " <texto>

**<respuesta>** ::= "A: " <razonamiento> <respuesta-final>

**<razonamiento>** ::= <afirmación> ("." <afirmación>)\* "."

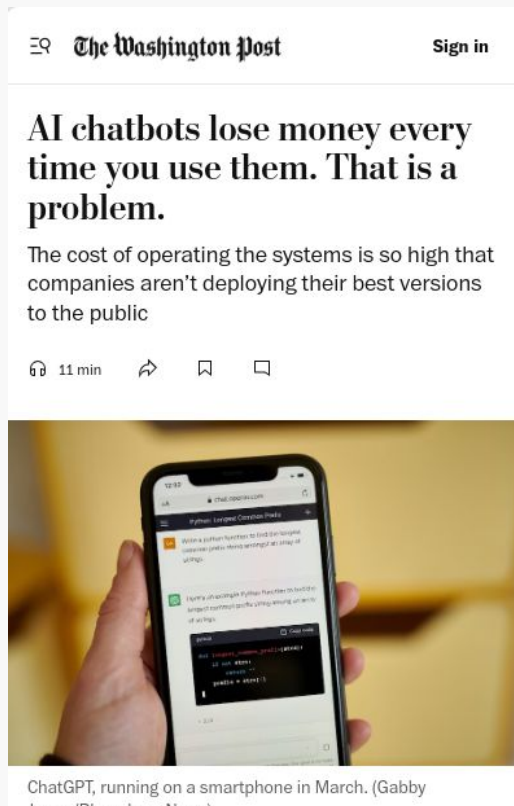
**<afirmación>** ::= <texto>

**<respuesta-final>** ::= "La respuesta es " <texto> "."



# Resumen

Crear una gramática formal para la generación de Prompts que luego evaluaremos formalmente utilizando un modelo probabilístico en PRISM con datos generados con el dataset de preguntas ARC. Además esto permitirá ahorrar costos operacionales a las empresas y mejorar las respuestas que obtienen los usuarios

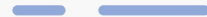






# 05 Trabajos relacionados

Sobre Chain-Of-Thoughts y Tree-Of-Thoughts





# Prompting Automático de Cadena de Pensamiento en Grandes Modelos de Lenguaje



## Resumen:

Propone Auto-CoT, un método que genera automáticamente cadenas de razonamiento para demostraciones en tareas de razonamiento. Utiliza diversidad para mitigar errores y logra un rendimiento comparable o superior a métodos manuales.



## Enlace:

<https://arxiv.org/abs/2210.03493>





# Grandes Modelos de Lenguaje son Razonadores de Cero-Shot



## Resumen:

Demuestra que añadir "Pensemos paso a paso" permite a los LLM realizar razonamiento complejo de cero-shot. Este enfoque, Zero-shot-CoT, supera significativamente el rendimiento de cero-shot en diversas tareas de razonamiento sin ejemplos manuales.



## Enlace:

<https://arxiv.org/abs/2205.11916>



# El Prompting de Cadena de Pensamiento Induce Razonamiento en Grandes Modelos de Lenguaje



## Resumen:

Explora cómo generar una cadena de pensamiento mejora el razonamiento complejo en LLM. El método de prompting con cadena de pensamiento, usando ejemplos de demostración, mejora el rendimiento en tareas de razonamiento aritmético, de sentido común y simbólico.



## Enlace:

<https://arxiv.org/abs/2201.11903>





06

# Código VDM

Revisar código de VDM



07

# Pruebas

Combinatorias y clásicas en VDM



# **Análisis formal basado en VDM++ de una gramática para mejorar el rendimiento cognitivo de los LLM**



# ⋮ En general...



## Mejora de resultados

Mejorar los prompts mediante gramáticas para mejorar el output generado por LLMS (Claude y ChatGPT)



## Control de comportamiento

Guiar a la IA en la mayoría de casos para que sea segura y evitar que el sistema de información no deseada



## Ahorro de recursos

Al tener respuestas más completas se reduce la necesidad de gastar tokens







# 08

## Diagrama de transición



# Introducción

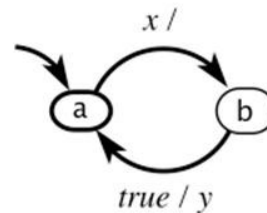
El modelo se diseñó con dos estados principales y variables asociadas que determinan las condiciones de transición. A continuación, describimos el modelo, su implementación en NuSMV y verificamos propiedades específicas mediante lógica temporal lineal (LTL). El sistema modela un flujo en el cual un usuario edita y valida un prompt (entrada).

Does the following hold?

$$G(x \implies Fy)$$

**input:**  $x$ : pure

**output:**  $y$ : pure



**yes**

```
MODULE main
VAR
    state : {editing, validating}; -- Estados del sistema
    isPromptValid : boolean;        -- Variable que indica si el prompt es
        v lido
    isPromptComplete : boolean;     -- Indica si el prompt est  completo
    promptRulesSatisfied : boolean; -- Reglas de validaci n

ASSIGN
    -- Estado inicial
    init(state) := editing;
    init(isPromptValid) := FALSE;
    init(isPromptComplete) := TRUE;
    init(promptRulesSatisfied) := TRUE;

    -- Transici n de estados
    next(state) :=
        case
            state = editing & isPromptComplete : validating;
            state = validating & !isPromptValid : editing;
            state = validating & isPromptValid : validating;
            TRUE : state;
        esac;

    -- Actualizaci n de isPromptValid
    next(isPromptValid) :=
        case
            state = validating : isPromptComplete & promptRulesSatisfied;
            TRUE : isPromptValid;
        esac;
```





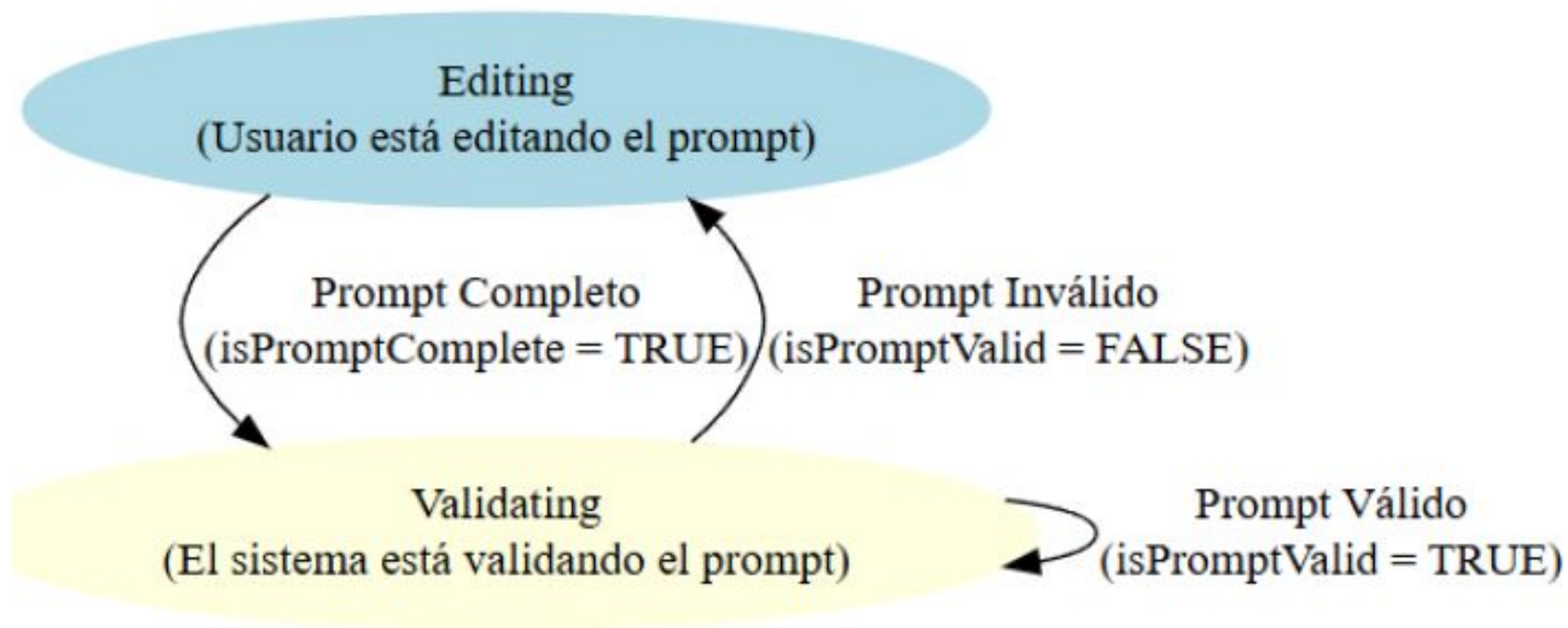
```
PS C:\Users\Alessandro\Downloads\NuSMV-2.6.0-win64\bin> .\NuSMV.exe -int
*** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:37:51 2015)
*** Enabled addons are: compass
*** For more information on NuSMV see <http://nusmv.fbk.eu>
*** or email to <nusmv-users@list.fbk.eu>.
*** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>

*** Copyright (c) 2010-2014, Fondazione Bruno Kessler

*** This version of NuSMV is linked to the CUDD library version 2.4.1
*** Copyright (c) 1995-2004, Regents of the University of Colorado

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://minisat.se/MiniSat.html
*** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
*** Copyright (c) 2007-2010, Niklas Sorensson

NuSMV > read_model -i prompt.smv
NuSMV > flatten_hierarchy
NuSMV > encode_variables
NuSMV > build_model
NuSMV > check_ltlspec
-- specification G (state = validating -> F state = editing) is true
NuSMV > |
```





09

# Conclusiones



# Conclusiones

**Este trabajo demuestra cómo el análisis formal basado en VDM++ puede mejorar significativamente la eficiencia y precisión de los modelos grandes de lenguaje (LLMs) mediante la estructuración y validación de prompts. La implementación de una gramática formal permitió:**

- Reducir el tiempo de procesamiento y el uso de recursos computacionales, optimizando las estructuras de los prompts.
- Incrementar la consistencia y calidad de las respuestas generadas por los LLMs, mejorando la adherencia a restricciones y formatos predefinidos.
- Validar de manera rigurosa las secuencias de tokens para garantizar la conformidad con las reglas gramaticales establecidas, alcanzando una tasa de detección del 95% para prompts inválidos.





10

# Referencias





# Referencias

Tema	URL
Bing	<a href="https://www.theverge.com/23599441/microsoft-bing-ai-sydney-secret-rules">https://www.theverge.com/23599441/microsoft-bing-ai-sydney-secret-rules</a>
Perdida de dinero	<a href="https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares">https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares</a> <a href="https://www.washingtonpost.com/technology/2023/06/05/chatgpt-hidden-cost-gpu-compute/">https://www.washingtonpost.com/technology/2023/06/05/chatgpt-hidden-cost-gpu-compute/</a>
Prompting	<a href="https://www.promptingguide.ai/techniques/cot">https://www.promptingguide.ai/techniques/cot</a>





**Gracias**

