ActionReasoningBench: Reasoning about Actions with and without Ramification Constraints

Divij Handa*[1], Pavel Dolin*[1], Shrinidhi Kumbhar*[1], Tran Cao Son[2], Chitta Baral[1]

[1]Arizona State University, USA
[2]New Mexico State University, USA
*Equal Contribution

Link - https://arxiv.org/abs/2406.04046

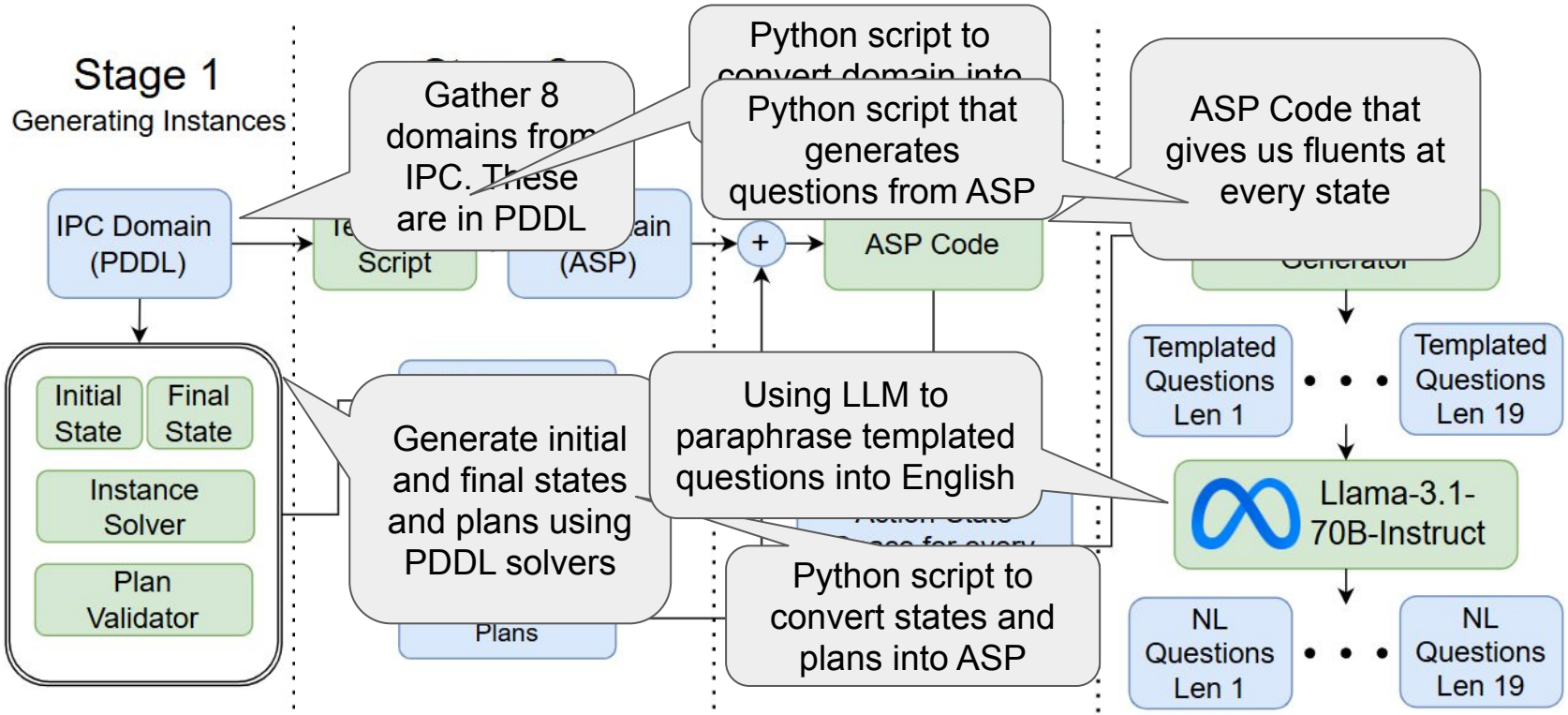
What is Reasoning about Actions?

- Reasoning about Actions and Change (RAC) is a fundamental problem in AI, where a system has to determine the effects of one or more actions
- An effect is defined as a change in the properties of an object, also known as **fluents** of an object
- Consider the statement: “Moving a ball from location X to location Y results in the ball being at location Y”
  - Action: Moving
  - Fluent/Property: Location
  - Object: Ball
  - Effect: Change in fluent from X to Y

Motivation

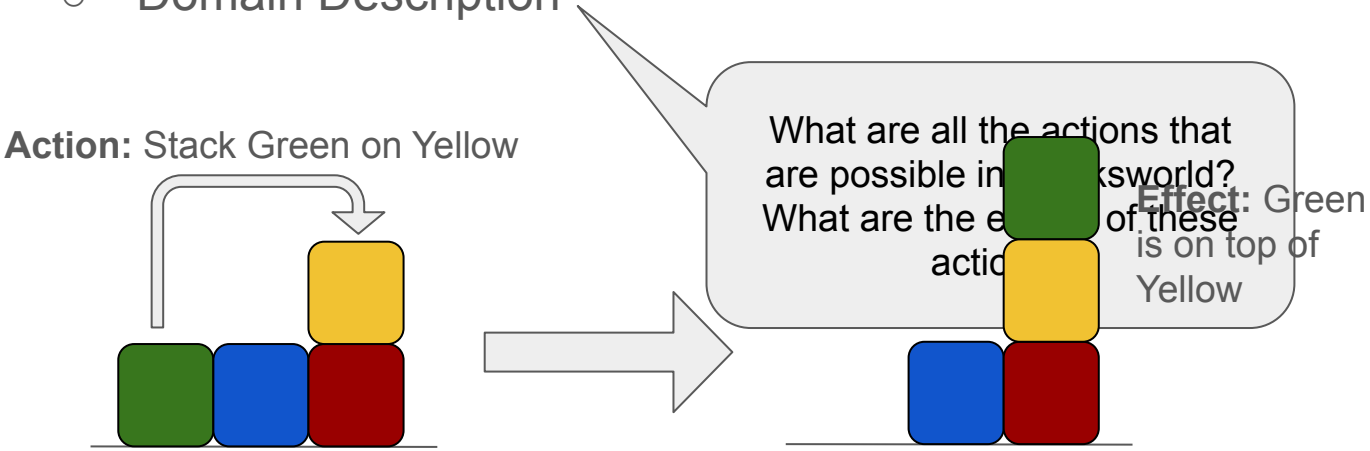
- Since LLMs are integrated as agents, can they reason over actions?
- Moreover, real-world actions are complex and can have indirect effects, known as **ramifications** of actions
- Limitations of past works:
  - No detailed analysis of where LLMs struggle while reasoning over their actions
  - Do not consider ramifications of actions, which is crucial for real-world systems
- Thus, we introduce ActionReasoningBench – a diagnostic benchmark for LLMs for RAC
  - Breaks Down RAC into six distinct categories which help in analysing failures
  - Introduce ramifications of actions with dependencies up to four levels deep, where actions propagate their effects through multiple layers

Data Creation Pipeline



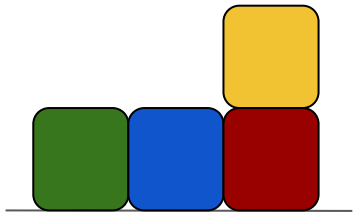
An instance of ActionReasoningBench

- Let’s consider an instance from one domain, Blocksworld
- Each instance of the dataset is divided into three sections:
  - Domain Description



An instance of ActionReasoningBench

- Let’s consider an instance from one domain, Blocksworld
- Each instance of the dataset is divided into three sections:
  - Domain Description
  - Initial Conditions

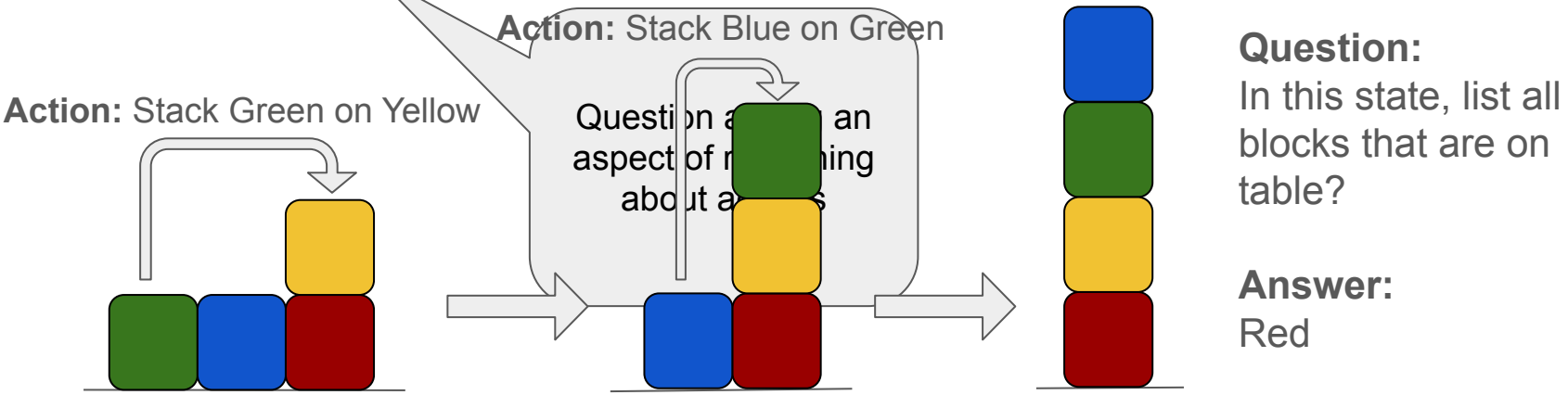


How are the blocks stacked in the initial state?

Green is on table, Yellow is on top of red, etc.

An instance of ActionReasoningBench

- Let’s consider an instance from one domain, Blocksworld
- Each instance of the dataset is divided into three sections:
  - Domain Description
  - Initial Conditions
  - Question



Question:
In this state, list all blocks that are on table?

Answer:
Red

An instance of ActionReasoningBench

[DOMAIN DESCRIPTION]
A block can only be picked up if it is clear, on the table, and the hand is empty, resulting in the block being held. A held block can be put down, placing it back on the table. Blocks can be stacked if the first block is held and the second block is clear, causing the first block to rest on top of the second. Unstacking occurs when the hand is empty, the first block is clear, and on top of the second, resulting in the first block being held again. A block can't be at two locations at the same time and is considered **clear** if nothing is on top of it and it's not held, and the hand is **empty** if it's not holding anything. Blocks are **stable** when clear and on the table, and they can be **painted** if stable and the hand is empty. A block is considered on **display** if it can be painted and has no other block on top of it.

[INITIAL CONDITIONS]
Block b1 is situated on the table, block b2 is not stacked with any other block, block b2 is also on the table, block b3 is not stacked with any other block, block b3 is positioned on top of block b7, block b4 is stacked on top of block b1, block b5 is not stacked with any other block, block b5 is placed on top of block b4, block b6 is on the table, block b7 is stacked on top of block b6, and the hand is empty.

[QUESTION]
Starting from the initial condition, the following actions are taken: block b3 is unstacked from the top of block b7 to achieve the current state. In this state, what are the valid properties (including both affirmative and negated properties) for b7? If there are no valid properties, write None.

About ActionReasoningBench

- Our benchmark contains 3,498 and 149,237 questions in the test and training sets respectively
- Questions contains up to 19 actions
- Both binary (True or False) and free-answer questions
- We categorized the questions into fundamental (first four) and complex (last two) questions
  - Fluent Tracking
  - State Tracking
  - Action Executability
  - Effects of Action
  - Numerical RAC
  - Composite Questions

Fluent & State Tracking

- Given the initial state and the sequence of actions performed, fluent tracking asks questions about the fluents, i.e. properties of the domain, of an object from the final state
- Example: *List down all properties of the Yellow block*

- Given the initial state and the sequence of actions performed, state tracking asks questions about all the fluents, i.e. properties of the domain, from the final state
- Example: *What are all the valid properties in this state?*

Action Executability

- This category encompasses two types of questions related to executability of actions
  - Given an initial state, and a sequence of actions, we randomly change one action to an action that is not possible in that state. Now we ask the LLM to *identify the first action in the sequence that is not executable*
  - Given an initial state and a sequence of actions leading to a final state, the task is to identify all actions that can be executed in the final state. *List all actions that can be executed in the current state*

Effects of Actions

- This category contains questions that explore the outcomes of performing a specific action from the given state.
- Example: *In the current state, if I unstack Yellow block from top of Red block, what properties of the state are true now?*

Numerical RAC

- These questions require a numerical response and are derived from the previously mentioned categories
- Example from Action Executability: *In the current state, what are the number of executable actions?*
- Example from Fluent Tracking: *In the current state, how many blocks are on the table?*
- Example from State Tracking: *How many properties of the state are false?*

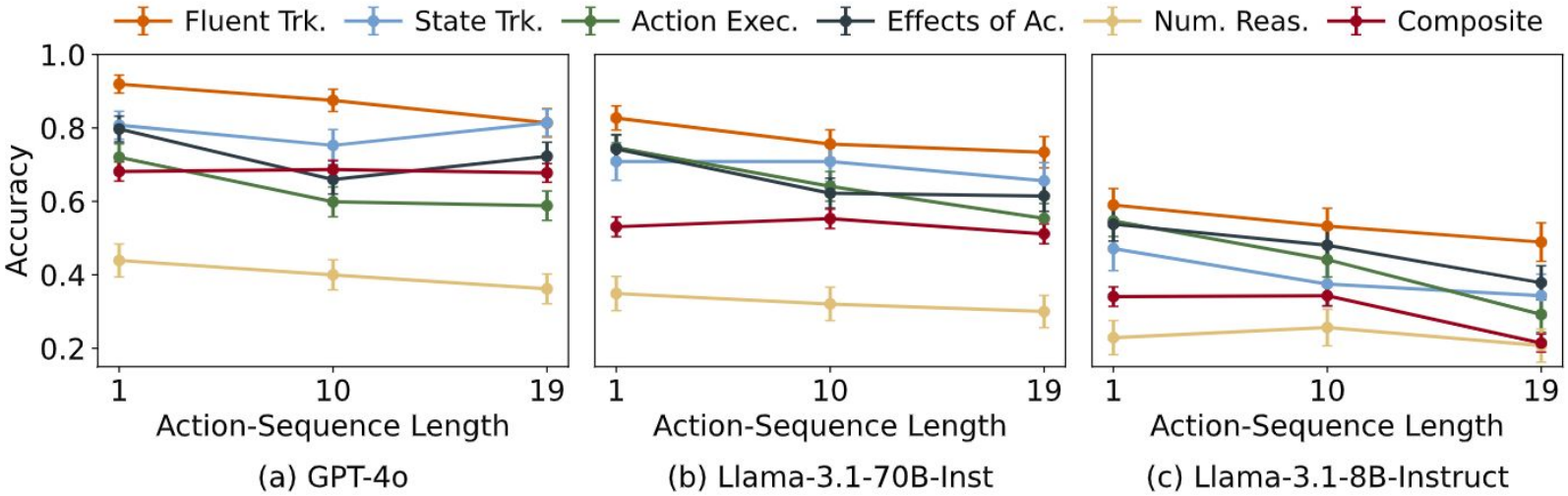
Composite Question

- Questions that integrate multiple categories, combining up to three distinct categories make up composite questions
- These questions require multiple steps of reasoning to arrive at the correct solution
- Example of Composite Question containing Fluent Tracking and Action Executability: *List all the properties of the state for Yellow block before the first infeasible action in the sequence?*

Experimental Setup

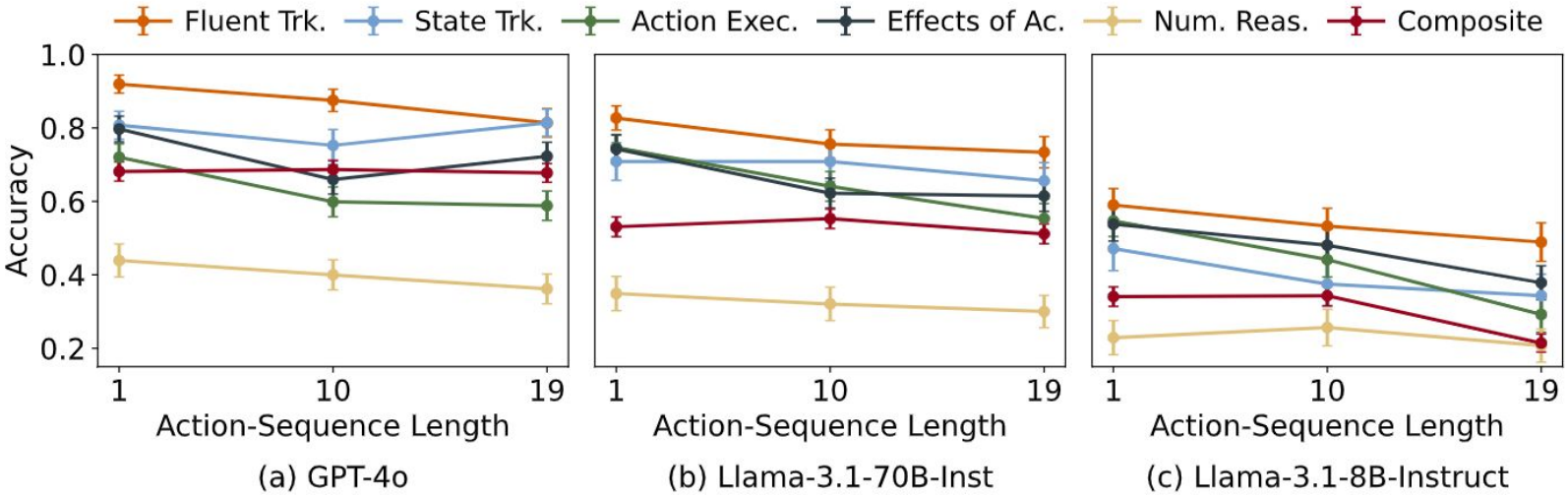
- Models
  - Llama-3.1-8B-Instruct
  - Llama-3.1-70B-Instruct
  - GPT-4o
  - o1-preview (only on the ramification subset, due to cost)
- Evaluation
  - For binary questions (True or False), we perform exact match
  - For free-response, we use LLM-as-a-judge approach

Results on Non-Ramification Subset



- Models struggle as action sequences get longer
- LLMs perform well in Fluent Tracking, State Tracking, and Effects of Actions, demonstrating their strength in keeping track of changes

Results on Non-Ramification Subset



- However, they struggle with Action Executability, Composite Questions, and Numerical RAC
- Bigger LLMs perform better
- LLMs struggle when asked about fluents that are false compared to fluents that are true (12.16% decrease)

Results on Ramification Subset

Action Sequence	Question Categories	Free Answer		Binary Questions	
		GPT-4o	o1-preview	GPT-4o	o1-preview
1	Fluent Tracking	$00.00_{00.00}$	$00.00_{00.00}$	$71.43_{17.10}$	$100.00_{00.00}$
	State Tracking	$00.00_{00.00}$	$20.00_{17.89}$	$100.00_{00.00}$	$100.00_{00.00}$
	Effects of Actions	$00.00_{00.00}$	$40.00_{21.91}$	$71.43_{17.10}$	$57.14_{18.70}$
	Average	$00.00_{00.00}$	$25.00_{12.49}$	$80.95_{08.57}$	$85.71_{07.60}$
10	Fluent Tracking	$00.00_{00.00}$	$00.00_{00.00}$	$57.14_{18.70}$	$57.14_{18.70}$
	State Tracking	$00.00_{00.00}$	$33.33_{19.24}$	$42.86_{18.70}$	-
	Effects of Actions	$00.00_{00.00}$	$14.28_{13.22}$	$71.43_{17.10}$	$100.00_{00.00}$
	Average	$00.00_{00.00}$	$23.07_{11.68}$	$57.14_{10.80}$	$78.57_{10.90}$
19	Fluent Tracking	$00.00_{00.00}$	$33.33_{27.21}$	$42.86_{18.70}$	$57.14_{18.70}$
	State Tracking	$00.00_{00.00}$	$00.00_{00.00}$	$57.14_{18.70}$	-
	Effects of Actions	$00.00_{00.00}$	$00.00_{00.00}$	$71.43_{17.10}$	$85.71_{13.20}$
	Average	$00.00_{00.00}$	$07.69_{07.38}$	$57.14_{10.80}$	$71.43_{12.10}$

- LLMs struggle with ramifications
- Upon manual inspection, we found that at each step, GPT-4o only considers the direct effect of actions, skipping over ramifications

Conclusion

- Introduced **ActionReasoningBench**, a **diagnostic benchmark** for evaluating LLMs on reasoning about actions
- Created two novel complex question categories – **Numerical RAC** and **Composite Questions**
- Introduced indirect effect of actions, known as **ramifications**
- Found that LLMs struggle with ramifications

Thank You!

Our Lab

Paper Link:
https://arxiv.org/abs/2406.04046

Full Paper