FixMyPlan: Leveraging Large Language Models to Fix Ill-Defined Models and Incorrect Plans

Marcus Tantakoun[1], Christian Muise[2], Xiaodan Zhu[3]
School of Computing, Queen's University, Kingston, Canada

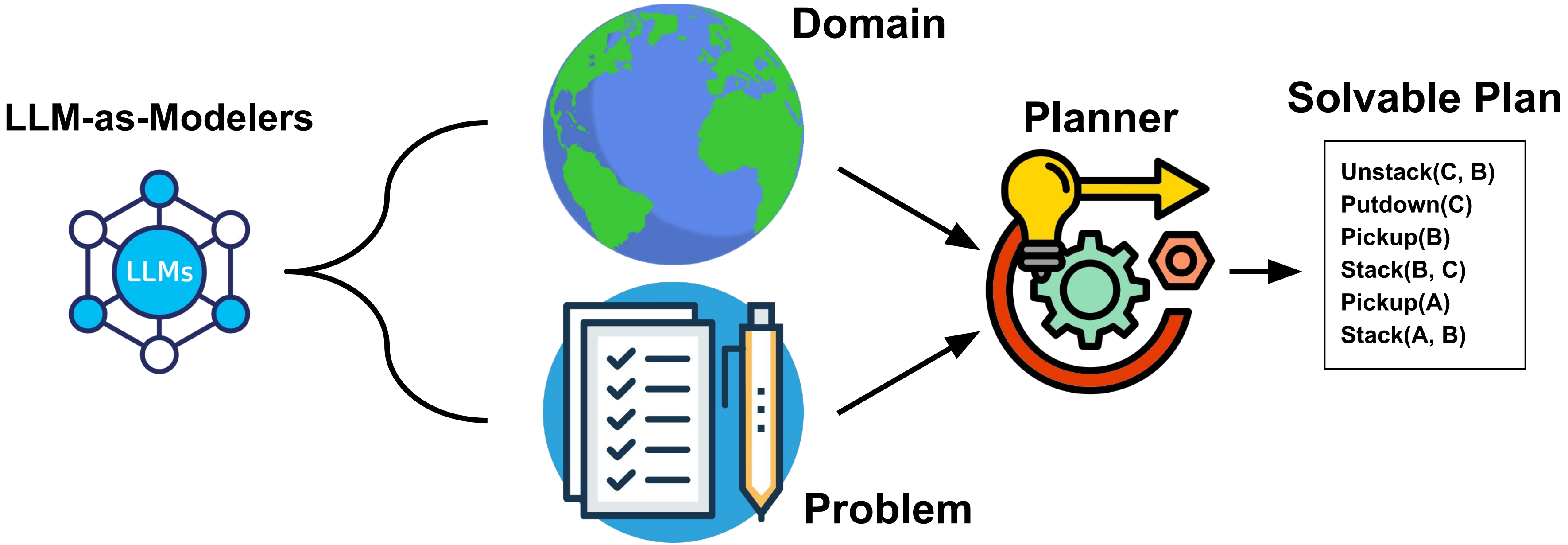
Mu Lab
at Queen's University

INGENUITY LABS
RESEARCH INSTITUTE
at Queen's University

Queen's
UNIVERSITY

FRAMING THE PROBLEM

LLMs have made an impressive strides in generating formal representations such as PDDL:



LLM-as-Modelers

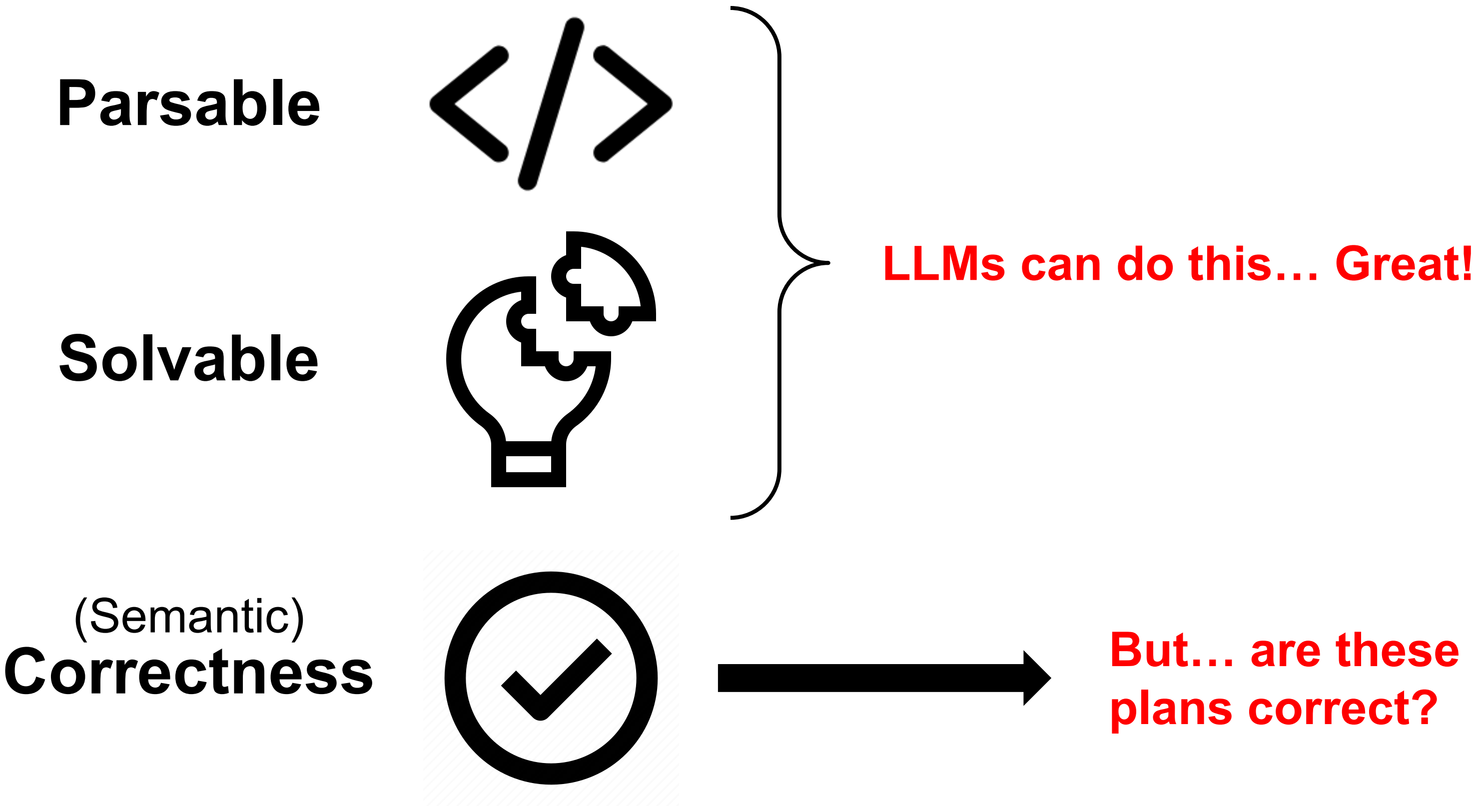
Domain

Problem

Planner

Solvable Plan

Unstack(C, B)
Putdown(C)
Pickup(B)
Stack(B, C)
Pickup(A)
Stack(A, B)



Parsable

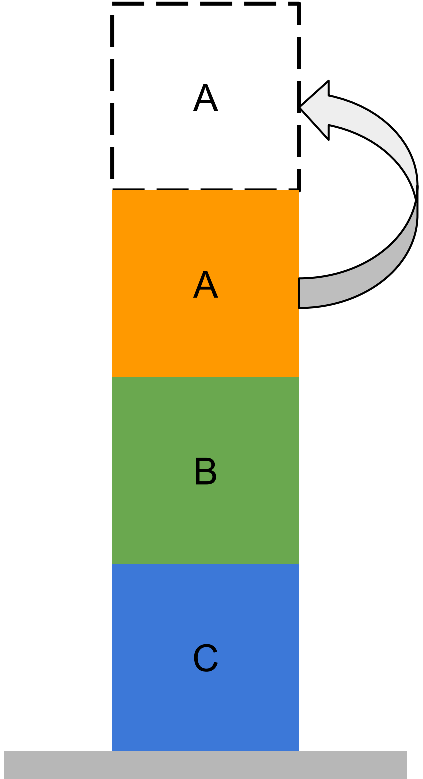
Solvable

(Semantic) **Correctness**

LLMs can do this… Great!

But… are these plans correct?

HUH?



Generated Plan Sequence

Step One

●
●
●

Stack(A, A) – Place **Block A** on top of **Block A.**

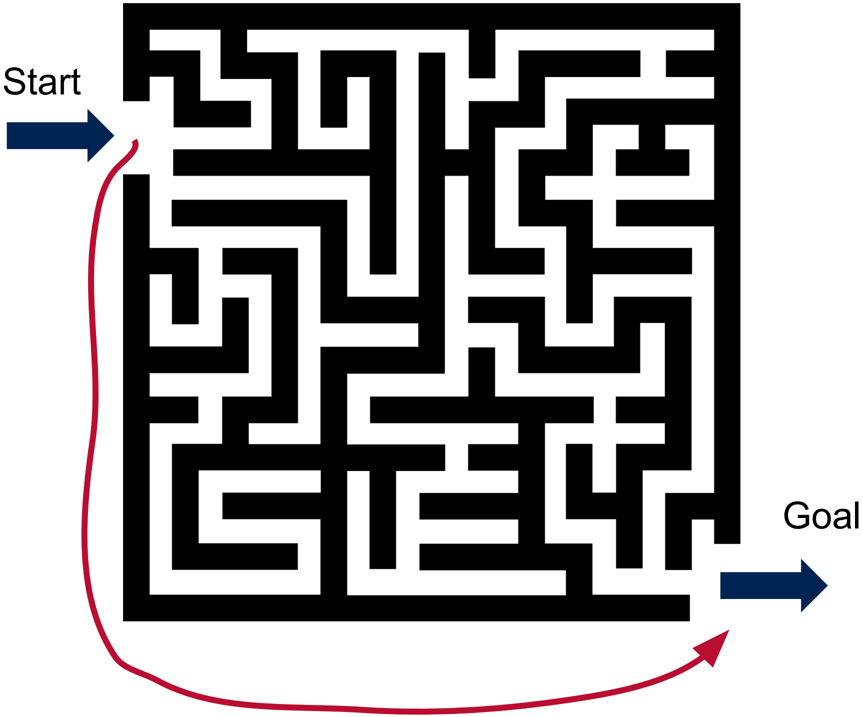
●
●
●

Step N

PROBLEM:
Plan output quality is **heavily dependent** on planning model quality

Even if we derive solvable plans from our planning models, **if the plans are not coherent, then what is the point?**

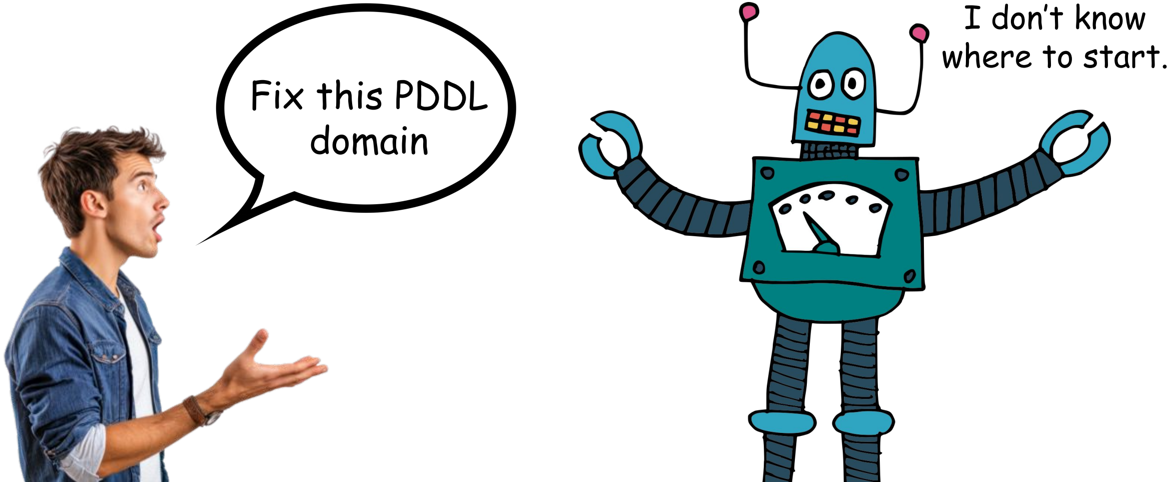
oops…



MOTIVATION

Prior research indicates LLMs struggle to directly correct semantic errors in PDDL domains (Gragera and Pozanco 2023; Patil 2024), **highlighting need for alternative approaches**

Adopting post-hoc approach, using generated plans as source of reliable feedback mechanism, enabling LLMs to detect and refine domain modeling errors more effectively



SOLUTION: FixMyPlan

Very reminiscent to LLM-Modulo approach!

→ First post-hoc approach of refining domains via plan analyzation

→ closed-loop framework that breaks down generated plan to detect semantic inconsistencies to fix underlying domain

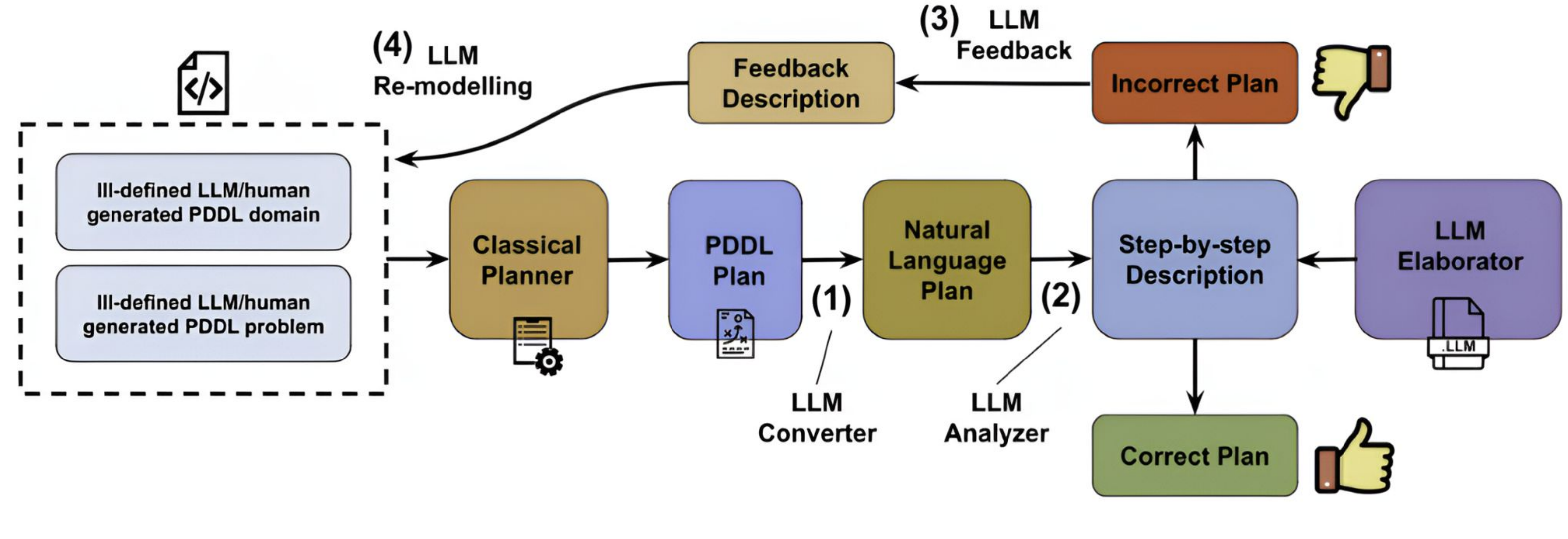
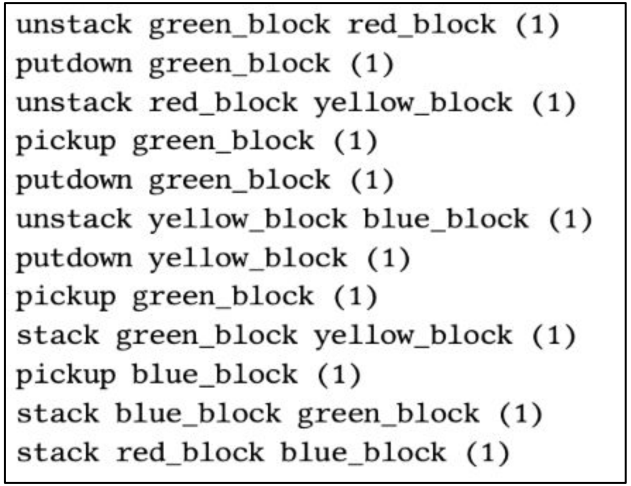


Figure 1: FixMyPlan general framework

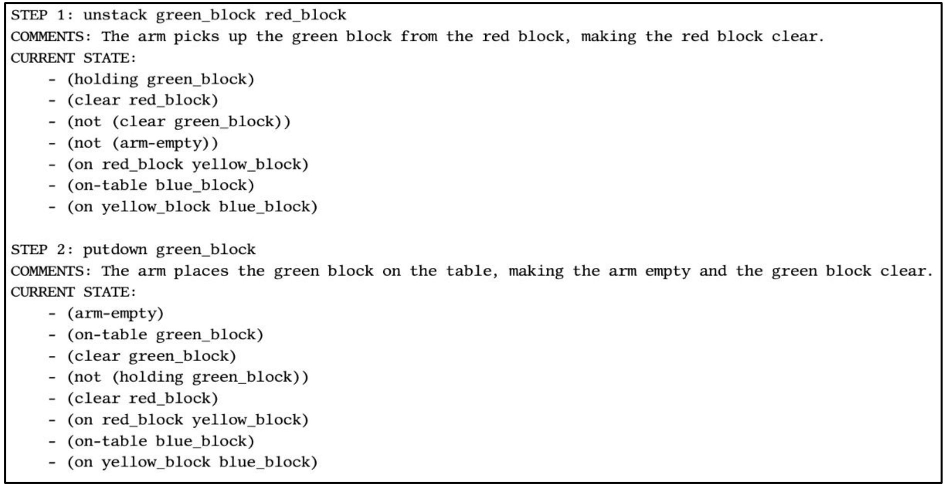
Example:

Original Flawed Plan



```
unstack green_block red_block (1)
putdown green_block (1)
unstack red_block yellow_block (1)
pickup green_block (1)
putdown green_block (1)
unstack yellow_block blue_block (1)
putdown yellow_block (1)
pickup green_block (1)
stack green_block yellow_block (1)
pickup blue_block (1)
stack blue_block green_block (1)
stack red_block blue_block (1)
```

Translated NL Plan (via LLM)



```
STEP 1: unstack green_block red_block
COMMENTS: The arm picks up the green block from the red block, making the red block clear.
CURRENT STATE:
    - (holding green_block)
    - (clear red_block)
    - (not (clear green_block))
    - (not (arm-empty))
    - (on red_block yellow_block)
    - (on-table blue_block)
    - (on yellow_block blue_block)

STEP 2: putdown green_block
COMMENTS: The arm places the green block on the table, making the arm empty and the green block clear.
CURRENT STATE:
    - (arm-empty)
    - (on-table green_block)
    - (clear green_block)
    - (not (holding green_block))
    - (clear red_block)
    - (on red_block yellow_block)
    - (on-table blue_block)
    - (on yellow_block blue_block)
```

Promising Outlooks:

- FixMyPlan pipeline performs **considerably on par with the baseline** at detecting and making correct domain fix
- Mystery domain showcases how **FixMyPlan enables LLMs to systematically analyze plan steps** through granular inspection, rather than **relying on retrieving from training corpus.**

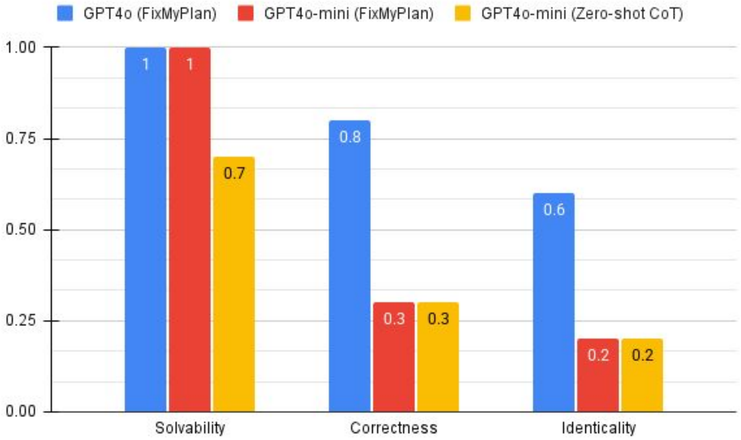


Figure 2: Mystery Logistics Domain. *Correctness* = validating plan via ground truth domain (VAL). *Identicality* = plan generated same as ground truth

CONCLUSION

FixMyPlan is a work in progress!

- Robust method to **tracking current predicate** states rather than tasking LLM
- **More efficient pipeline** — reuse of LLM translating (step 1) and LLM analyzer (step 2)
- **Increasing test dataset**

FixMyPlan is the first work to explore **post-hoc methods** for **identifying and correcting semantically incorrect plans** (In lens of LLM-as-Modelers)

→ Further research into addressing the **semantic integrity of PDDL model generation through LLMs, not only by detecting flawed plans but also by improving the underlying domain modeling process.**

Any Questions?