Explainable Gridworld Navigation Using Prolog

Ayush Salunke
Ayushi Arora

Mentor: Youssef Mahmoud Youssef

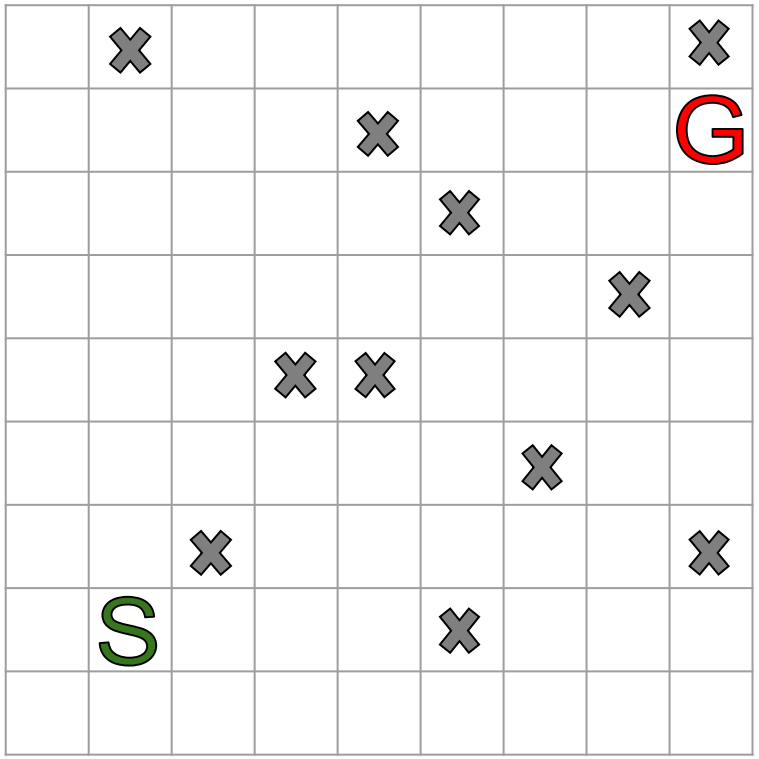
Problem Statement

This project builds an explainable Gridworld agent.

- Need for transparent decision-making in AI systems
- Grid navigation as a simple yet representative problem
- Black-box models lack interpretability
- Goal: build an explainable, rule-based navigation system
- Objective: Reach the goal while avoiding obstacles.

Gridworld environment

- 9 × 9 grid
- Start position: (0,0)
- Goal position: (8,7)
- Obstacles placed manually
- Allowed actions: up, down, left, right

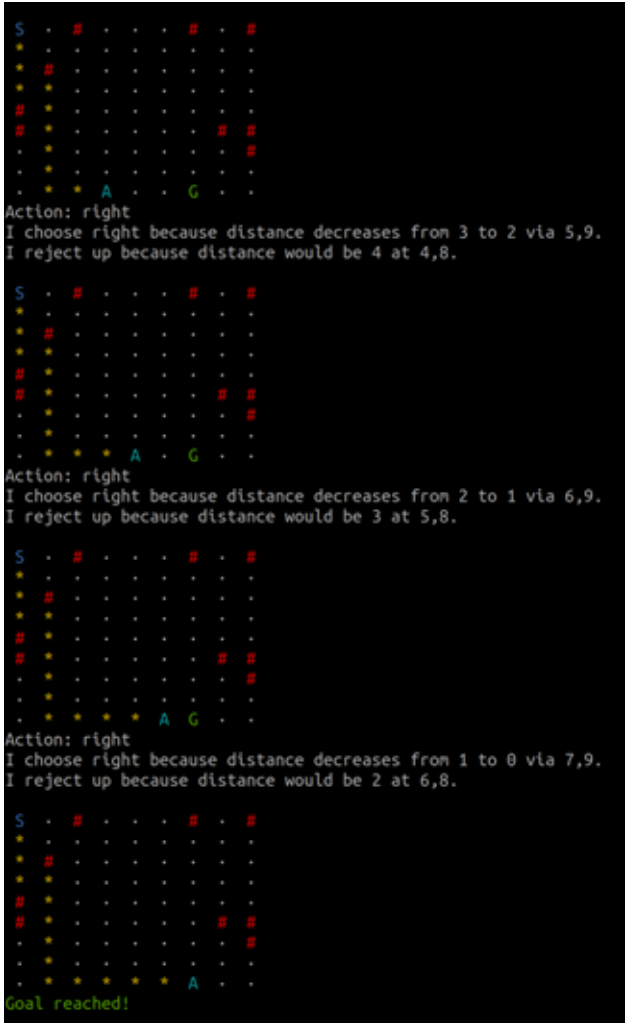


Logic Representation and Inference

- Environment knowledge encoded as logical facts
- Constraints define valid and invalid moves
- Goal-directed reasoning guides action choice
- Inference engine derives actions step by step

Explainable Decision Making

- Actions evaluated symbolically
- Each action justified by rules
- Distance-based reasoning toward goal (Manhattan Distance)
- Human-readable explanations generated
- For example: The agent provides an simple explanation for each action that it takes and also provides an explanation for the action that it rejects based on a simple heuristic distance from goal.



Verification: Positive & Negative examples

- Positive examples define correct behavior
- Negative examples define forbidden actions
- Automated testing in Prolog
- Logical consistency is verified
- For example:

```
?- best_action((8,5), up).
true.

?- best_action((6,4), right).
true.

?- best_action((4,2), right).
true.
```

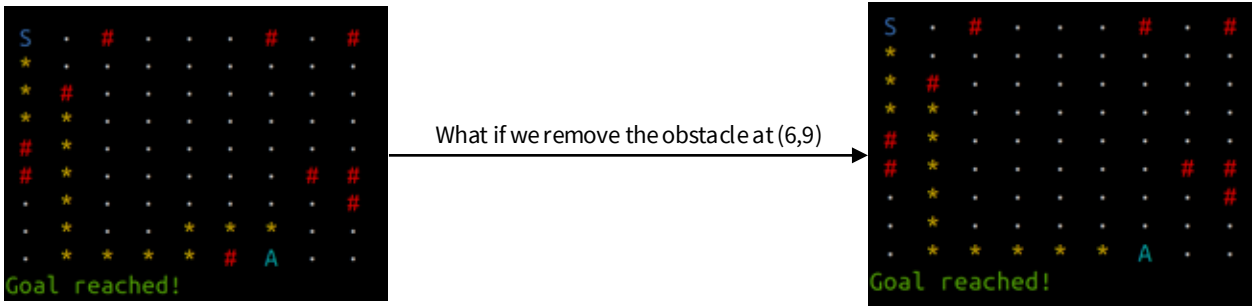
Positive Example

```
?- best_action((6,3), up).
false.

?- best_action((9,5), up).
false.

?- best_action((0,0), left).
false.
```

Negative Example

Counterfactual Reasoning

- This is a simple “What-if?” analysis.
- It helps enable us a causal reasoning like, “If a certain fact is changed in the world, how would it affect the output?”
- Supported counterfactuals for our program are:
  - What if we remove an obstacle
  - What if we add an obstacle
  - What if we change the goal
  - What if we change the start position



*Note: for now we have just implemented this in the random gridworld program only for testing purposes

Limitations & Future Work

Limitations:-

- Currently there are no global planning algorithm
- The current greedy heuristic (manhattan distance) is not an optimal algorithm
- Also there is no learning from experience with our program.

Future work:-

- We can work on a different search algorithm like A*.
- We can allow diagonal movements for the agent.
- We can also implement a bigger gridsearch or provide a real time UI to track the agents movements.

Thank You!

Any Questions?