Project Milestone 3: Progress Report on Automated Warehouse Scenario using ASP

Poorvik Dharmendra
School of Computing & Augmented Intelligence
Arizona State University
pdharmen@asu.edu

Abstract

This report delineates the progress made in the Automated Warehouse Scenario project for CSE 579 at Arizona State University. It succinctly introduces the problem statement, provides a summary of the progress made to date, discusses encountered issues along with the proposed action plan, and concludes with a summary of completed tasks and those slated for completion in the coming days.

Problem Statement

In an automated warehouse, robots operate within a rectangular grid to transport shelves that contain specific products to picking stations for fulfilling various orders. These robots are capable of horizontal or vertical movement between adjacent cells and possess a flat design that allows them to maneuver beneath shelves. However, it's important to note that a robot carrying a shelf cannot pass beneath another shelf, thus requiring repositioning of shelves to allow specific movements. The primary objective is to achieve efficient order fulfillment, measuring time in steps, with each robot having to perform one action per step, movement, shelf handling, or remaining idle. Collisions are strictly prohibited, ensuring no two robots occupy the same space at the same time. Additionally, there are designated highway cells where robots cannot place shelves.

The project requires the use of Answer Set Programming (ASP) to formulate hard constraints that will determine a strategic course of action for the efficient fulfillment of all orders within the minimum possible number of time steps.

Summary of Progress

As part of the course curriculum, the project was divided into four milestones – introduction to Clingo, followed by writing basic programs in Clingo, and a dedicated progress report for Automated Warehouse Scenario project. The final milestone culminated in the submission of the completed project for the Automated Warehouse Scenario.

Dr. Joohyung Lee's course lectures and slides played a pivotal role in my understanding of solving various problems using ASP. Module-3 introduced me propositional rules and programs. Building upon this, Module-4 delved into the concepts of constraints and cardinality, offering insight into writing effective constraints with practical applications demonstrated through scenarios like the N-Queen and N-Bishops problems.

Equipped with this basic knowledge, I studied the project description provided with the kit. I carefully iterated through the problem statement and various deliverables associated with it. I broke down the problem into separate sections, each tackling a specific goal. For instance, one section that focused purely on setting up the grid cells, and another section dedicated to defining the movement conditions and constraints. I jotted down the identified conditions and constraints in the natural language.

I successfully wrote propositional rules to initialize various environment parameters such as number of rows and columns in the grid, number of shelves, picking stations, orders, and robots, by parsing the object initialization programs provided with the kit. The next step would be converting the various movement constraints defined in natural language to propositional programs.

Issues & Challenges Encountered

The foremost challenge I encountered was the uncertainty surrounding the identification of all necessary constraints required to chart a viable course of action. It was difficult to check if the set of conditions were robust as it was expressed in natural language without a tangible Clingo program for testing them.

Secondly, as I delved into writing the initialization rules, it was challenging to determine whether the rules were appropriate for the problem statement. Ensuring that they generated all required objects in accordance with the problem's specifications posed a significant challenge.

Moreover, I found it extremely challenging to progress with the code in the absence of any guarantee that the written conditions indeed led to an optimal solution, behaving as intended. Without completing the project, it was difficult to erase these doubts, which significantly hindered my ability to progress. Additionally, translating conditions from natural language to precise constraints and propositional rules in Clingo posed its own set of challenges. A subtle shift in the placement of constraints, be it in the head or body of a propositional rule, resulted a substantial impact on its interpretation.

Plan to Resolve Issues

To navigate these challenges, I adopted a systematic approach. First and foremost, I understood that transitioning from natural language conditions to testable Clingo program involves constant iteration. I will consider a very small scenario with a few robot workers and a few grid cells to see how the conditions behave and evolve the rules from there. Additionally, seeking help from peers and reading online resources will provide valuable insights, usually offering a fresh perspective on potential oversights.

The above approach will also help in validating the initialization rules. I will continuously review the rules against the problem statement to ensure alignment with the requirements. Finally, my approach of breaking the problem into sections will help address doubts about the optimal solution. Gradual, step-by-step testing and validation of smaller components can help build confidence in the overall solution, allowing for adjustments and refinements as needed.

Tasks Completed

- Installed and configured Clingo on local system.
- Studied basics of Answer Set Programming from course lectures and slides.
- Completed practice of Answer Set Programming from Module-4 and learnt how to write rules and constraints.
- Understood Reasoning about Actions and Common-Sense Law of Inertia from Module-5.
- Practiced writing programs in Clingo and solved problems such as N-Queens, Blocks World, Monkey & Banana.
- Downloaded the project kit and initializations source code from Canvas.
- Studied the project description and listed down the deliverables.
- Divided the project into different sections and noted down the conditions and constraints in natural language for each section.
- Wrote environment variable initialization rules in Clingo to determine number of robots, cells, orders, picking stations and so on.
- Identified Common Sense Law of Inertia constraints for the locations of all objects in the Automated Warehouse Scenario environment.

Future Tasks and Plans

- Finalize the formulation of the remaining constraints and rules within each identified section to comprehensively address the project's requirements.
- Establish a small-scale test case scenario with a limited number of objects to rigorously test the written conditions, facilitating necessary improvements.
- Evaluate all potential actions in the scenario and ensure their conformity to the already established Common-Sense laws of Inertia.
- Systematically test the program's functionality and its efficiency.
- Implement optimizations to the program, aiming to generate a plan to deliver all orders with the minimum possible time steps.
- Develop a robust framework to test the optimized solution proposed by the program.
- Compile a comprehensive report to elucidate the undertaken work to complete the project, substantiated by relevant proofs and supporting documents.
- Submit the report and program for evaluation.

References

Dr. Joohyung Lee. CSE 579 Lecture Videos, School of Computing and Augmented Intelligence, Arizona State University., Tempe, Arizona.

Dr. Joohyung Lee. CSE 579 Lecture Slides, School of Computing and Augmented Intelligence, Arizona State University., Tempe, Arizona.