Using JSHOP External Calls to Compute Geometric Predicates in a Hierarchical Task Network Planning

Guilherme V. B. de Azevedo
Programa de Pós-Graduação em Ciência da Computação
Pontifícia Universidade Católica do Rio Grande do Sul

Abstract

] The gap between symbolic and geometric planning is something that limits the usability of planning for robotics. In the last years some solutions have been developed to create link between these two areas. The idea of this project is to try to connect a HTN planner to an API that works with geometric predicates. The resultant tool will be evaluated on a simple 2D domain involving moving a robot inside a room.

Introduction

Symbolic planners have been improved drastically its ability to solve complex problems. However, due to the lack of correlation between the physical world and the symbolic predicates, they use in robotics are limited (Cambon, Gravot, and Alami 2004). For example, the shape or size of a robot, or even its new configuration after a grasp action are generally not translated into symbolic predicates. Geometric planning involves motion planning, collision-free trajectory, grasping poses among others problems that are related to the robots orientation on the 3 dimensional world (de Silva and Meneguzzi 2015).

Our idea in this project is to use HTN to describe a domain that handle symbolic and geometric predicates. JSHOP can handle numeric predicates. JSHOP also provides access to external classes using the external calls, which we intend to use to call an API that generates spatial predicates for geometry process called JTS Topology Suite, this way, allowing our domain to reason about the geometric configuration of a robot.

(Srivastava et al. 2013) mixed classical symbolic planner with geometric planner, synchronizing the discrete and continuous layers using representations techniques from first-order logic. Their work assume that each high-level action correspond to a continuous implementation, and they focus on representing these continuous operators in high-level, discrete planners. (Cambon, Gravot, and Alami 2003) developed AsyMov, an integration of motion, manipulation and task planning. They use PDDL2.1 and define a series of symbolic predicates as the minimal set of predicates for a basic geometric problem and the linking to the geometric space. They keep at each state a symbolic and geometric description of the robot configuration, providing more flexibility and allowing progressive refinement of the plan.

Technical Approach

This projects aims to develop an initial work on creating a symbolic-geometric planner. For such, we will explore the external calls of JSHOP to invoke an API for geometric processing class in order to generate a geometric plan. Despite the idea is to develop a solution that is domain independent, we will create a simple domain to evaluate the results of our solution that is further describe in Section . The motive of a simple domain is that we do not know how difficult is to use the JTS API, neither how complex can be the external calls made from JSHOP, if this task were easier that we are considering, a few improvements on the domain will be made, like: replacing the robot for a non-holonomic one, adding obstacles in the middle of the room or creating more tasks to the robot complete. Otherwise, if the external calls represent a bottleneck on the project, we may have to implement a wrapper that connects both JSHOP and the geometric suite. If such connection show not as simple as we are thinking, possible some JSHOP source codes will need to be edited, increasing the complexity of the solution.

Domain description

For this problem, two domain descriptions need to be created, one in HTN, with the symbolic predicates, and other in the API with the geometric objects. An equivalence between the two is necessary so, the predicates need to be constructed according to the informations that we need to extract from the real world. The JTS suite can provide informations (e.g. if there is an intersection between the geometry of two objects, calculate the centroid of an object, distance between points, area of an object) that we will need to use in our symbolic representation to generate the plan to make our robot to enter the charging station correctly. Each object is defined by a list of coordinates on the 2 dimensional Euclidean space, so it can have the most variable shape. We chose to represent a robot as a square and the charging station as an rectangle since the domain is not the focus of this project, but the properly connection between the two tools.

JTS Topology Suite

JTS topology is an open source JAVA library that provides an object model for Euclidean planar linear geometry with a set o geometric functions. An overview of the tool can be found at this presentation[1] from Martin Davis. The JTS Topology suite has tools to describe geometric objects from points in space to complex objects composed by multiple polygons. It provides predicates such as *intersect, touches, overlaps* that provides to the symbolic planner informations about the objects configurations. It also has some operations to calculate length, area distances intersections.
JTS is implemented in Java, but it has c++ (GEOS), c# (Net Topology Suite) and JSTS (Javascript) implementations. We chose the Java version because of that JSHOP is also implemented in Java and seems to be easier to make a link between them.

Project Management

The first step for the development of this project is to study the JTS suite and implement a few simple programs using the tool. A similar initial effort will be needed for understand the capabilities and limitations of the external calls in JSHOP. After this first stage, a study about ways to connect these two will be made, there is a chance that the external calls are not enough and, in this case, another tool wrapping both JSHOP and JTS will need to be constructed, possibly meaning some changes into the JSHOP source code. The next two steps are probably the less complicated ones, which consists of developing the domain for the problem. That is, creating geometric objects representing the robot, the charging station and the room, as well as symbolic predicates representing the necessary configurations to build a link between the two representations of world. At the end of the project, we will write a report about the experiments and results from the project, together with a presentation to present this project in class.

Schedule

- Study of JTS suite - week 1
- Study of JSHOP external calls - weeks 2 and 3
- Develop the connection between the geometric suite and the HTN planner - weeks 3 and 4
- Describe the domain and evaluate the results - week 5
- Write the report and the presentation - week 6

Conclusion

The idea of the project presented in this assignment is to create a tool to connect two layers of planning. This problem have many interesting uses such: pick and place problems, navigation and personal assistant robots. This project is the starting point to a masters degree thesis where we intend to approach not only this problem with more complexity but the generation of geometric predicates using sensors and planning in a partially observable and non-deterministic environment.

[1] $http : //tsusiatsoftware.net/jts/files/jts_{s}ecrets_{f}oss4g2007.pdf$

References

Cambon, S.; Gravot, F.; and Alami, R. 2003. Overview of asymov: Integrating motion, manipulation and task planning. In *Intl. Conf. on Automated Planning and Scheduling Doctoral Consortium*.

Cambon, S.; Gravot, F.; and Alami, R. 2004. A robot task planner that merges symbolic and geometric reasoning. In *ECAI*, volume 16, 895.

de Silva, L., and Meneguzzi, F. 2015. On the design of symbolic-geometric online planning systems. In *2015 Workshop on Hybrid Reasoning (HR 2015)*, 1–8.

Srivastava, S.; Riano, L.; Russell, S.; and Abbeel, P. 2013. Using classical planners for tasks with continuous operators in robotics. In *Intl. Conf. on Automated Planning and Scheduling*, 27–35.