Planning and Scheduling at home project

Ángela Patricia Enríquez Gómez, Ethan Oswald Massey, Maximilian Mensing

January 18, 2019

BRSU

Introduction

Task

Storing Groceries

- **Case 1**: Everything is known. One object is on the table and has to be placed at any shelf after the cupboard has been opened.
- **Case 2**: The amount of Objects, the table and the cupboard have to be located. The objects have to be placed on the shelves.
- **Case 3**: As case 2 - In addition to not knowing the number of objects, the objects themself are also unknown.
- **Case 4**: In addition to case 3 the cupboard is unknown and has to be explored. The items have to be put in different categories and sorted by category on the shelf.

Selection of the Planner

Using JSHOP2

- Get jshop from `https://github.com/mas-group/jshop2`
- Set environment **export CLASSPATH=”‘pwd‘/bin.build/JSHOP2.jar:‘pwd‘/antlr.jar:.”**
- Compile using **make c**
- Run by calling **make problem1/2/3/4**

Domain Overview

- Domain models include:
    - operators
    - methods
    - axioms

Domain Assumptions and Approach

Common sense assumptions were made to limit domain complexity.

Assumptions include, but are not limited to:

- Position of all objects are known and static (unless acted on by the robot)
- There is exactly the number of objects required (e.g. only one table)
- The robot is able to sense and identify all objects correctly

Testing began with the smallest domain (Problem 1) and gradually increase in complexity

Example - Problem 1

What’s in a Problem File

- Problem files are created for each problem
- Contain inital state and the compond task to be solved
- HTN planner uses this to create a solution

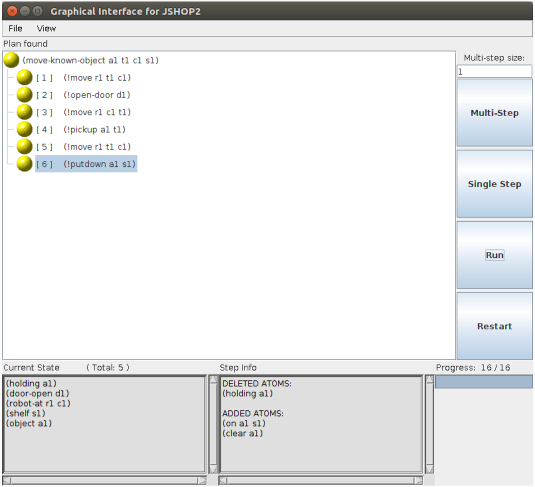
Problem Outline

- Location of the table and cupboard are known
- There is only one object on the table and it's location is known
- The cupboard door is closed
- The object can be placed on any shelf

Problem Definition

```
(defproblem problem1 storegroceries ;;Problem 1 ( (object a1)
(cupboard c1) (door d1) (shelf s1) (table t1) (robot r1) (on a1 t1)
(door-closed d1)(robot-at r1 t1) ) ((move-known-object a1 t1 c1
s1)) )
```

GUI Result



Plan found

(move-known-object a1 t1 c1 s1)
- [1] (!move r1 t1 c1)
- [2] (!open-door d1)
- [3] (!move r1 c1 t1)
- [4] (!pickup a1 t1)
- [5] (!move r1 t1 c1)
- [6] (!putdown a1 s1)

Multi-step size: 1

Current State (Total: 5)

```
(holding a1)
(door-open d1)
(robot-at r1 c1)
(shelf s1)
(object a1)
```

Step Info

```
DELETED ATOMS:
(holding a1)

ADDED ATOMS:
(on a1 s1)
(clear a1)
```

Progress: 16 / 16

Figure 1: GUI Problem 1

Limitations and Issues

Limitations

- Without sufficient prior information the planner is not able to classify objects
- If assumptions are invalidated the planner will fail
- In problem 4, if the shelves don't have example objects the planner has problems putting categories for them.
- Executing with java -ra generates all possible plans
  - High space and time complexity
  - Maximum of 4 objects to avoid outOfMemoryError

Diskussion

Offene Punkte

-