Shrdlite Project Report

Alexnder Sjosten **Asbjorn Hagalin** **Behrouz Talebi** **Runar Kristinsson**

Abstract

This document contains the instructions for preparing a camera-ready manuscript for the proceedings of EACL-2014. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the direction reported in this document.

Introduction

what is our main objective

Background

explain shrdlue, what is it and why is it important?

Interpretation

The interpreter will, given a parse tree from the parser, create a list of goals that should be given to the planner. Each goal will be a list of subgoals (a subgoal will be refered to as PDDL from here on out). When interpreting the the given parse tree from the parser, the interpreter module will check its interpretations (i.e. the list of goals) to the given world and rule out the interpretations that are impossible in the current world. If no interpretations are valid in the current world, there will be an interpretation error and if the list of goals contain more than one element there exists an ambiguity and it will be the job of the ambiguity resolution module to resolve the ambiguity.

Interpreting quantifiers

There exists three quantifiers, namely "the", "any" and "all" and the design choices of how to interpret each of the quantifiers will vary depending on the context.

The "the" quantifier

When interpreting the quantifier "the", it will be interpreted as a specific object. Assume you have the sentence "move the white ball to the red box". This sentence makes sence if it is uttered in a world that contains exactly one white ball and one red box. If there exists two or more white balls and/or two or more red boxes, it will be a situation where the ambiguity resolution will have to be called. Assume the utterance given above and a world containing two white balls (call them a and b) and one red box (call it c). The interpreter will in this situation create two goals, one that says a should be on top of c and one that says b should be on top of c.

The "any" quantifier

The "all" quantifier

Ambiguity resolution

Behrouz writes about how it's done.

Planner

From the beginning we wanted to create a planner that would make as few moves as possible to succeed at it's goal. We ended up creating Breadth first search, with heuristics that gives the algorithm lower bound and a specific node to go down. How does it work?

To find the lower bound we have a very simple cost function, it counts how many blocks are above the blocks that need to be moved to reach goal state. For example, if the goal is to put block a on top of block b, block a has 2 blocks on top of it and block b has 1 block on top of it. Then the lower bound would be (2+1) + 1 = 4. The reason for +1 is because we have to move a on top of b.

To find the specific nodes to start at we sort the possible moves in the world, so we first move blocks in the stacks that are used in the goal, starting in the stack with more blocks above block involved in the goal. For example if we use the example above then we would start with moves from the stack where block a is, because block a has more blocks on top of it, then we would move from the stack where block b is and then the rest of the moves

With these heuristics the algorithm uses Breadth first search, looking for goal state at the given lower bound first using most likely moves.

Conclusion

what have we learned, did we achieve the goals given in the introduction etc.

Acknowledgments

Do not number the acknowledgment section. Do not include this section when submitting your paper for review.

Appendix

What have each individual done.