An Agent-based Approach to Resolution Theorem Proving

Connor Langlois

Computer Science Department, University of Maine
Orono, ME 04469

connor.langlois@maine.edu

***Abstract* — This paper describes an approach to resolution theorem proving. It includes a write-up of the agent's behavior in the resolution theorem proving domain.**

***Keywords* — RTP: resolution theorem proving**

I. Introduction

The purpose of this paper is to describe the implementation of a resolution theorem prover. The prover takes as input a knowledge base and a theorem to prove. It proceeds by negating the inputted theorem and performing either Depth-First Search or A* to find a contradiction, which thus proves the theorem. If no such contradiction is possible (or cannot be found given the depth limit of 100,000) then the prover terminates without proving the statement.

II. Architecture

The main architecture of the prover is based on the simulator, world, agent, and agent program architecture as outlined in the paper *An Agent-based Approach to Robot World and N-Queens*. It contains a world that holds the knowledge base and an agent whose state is the leftover axiom and axiom resolved with to get that leftover axiom. The prover takes as input a knowledge base name, which represents a file that contains a list of axioms in conjunctive normal form (CNF) in LISP syntax, and a theorem to prove.

A. Control Strategies

1) *Set of Support*: Both Depth-First Search and A* employ the Set of Support control strategy. This strategy ensures that possible resolution states are only those that use the (negated) theorem trying to be proved. It is a good strategy as it is intuitive that to prove the theorem, the prover should resolve it with others to find a contradiction.

2) Unit Preference: Only A* employs the Unit Preference control strategy / heuristic. It follows the resolution state that results in a leftover axiom consisting of the fewest predicates of all the current possible resolution states. Thus, it is intuitive that this should allow A* to prove the theorem more quickly than Depth-First Search as it is greedy and results in fewer axioms to resolve.

III. Results

An interesting feature of the implementation is that it is based on the agent/simulator framework laid out in *An Agent-based Approach to Robot World and N-Queens*. Thus, the prover can be thought of as an agent / agent program, trying to use the negated theorem to prove the theorem with the axioms from the knowledge base, held by the world. After successfully finding a path to contradiction "in its head", the agent then uses the axiom used to resolve found in the agent state on the path to actually undergo the proving process. The simulator then checks to make sure the agent has indeed successfully proved the theorem.

From this, it is fairly easy to extend the prover due to the modularity and immutability of the framework. In successfully testing with Depth-First Search, it was only a matter of minutes to add A* to the mix.

One problem encountered was how to proceed with searching while keeping past states intact. This was overcome by utilizing the agent/simulator framework, which inherently leads to immutability of agent states between actions.

IV. Conclusions

Overall, it was found that the search techniques implemented performed well. A* did not necessarily perform better than Depth-First Search, however. This could be attributed to the fact that extremely simple knowledge bases were used and thus was an edge case where Depth-First Search could outperform A*.

Acknowledgment

Connor Langlois wishes to acknowledge Roy M. Turner for his teachings and guidance in Artificial Intelligence.

References

[1] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Upper Saddle River, New Jersey, United States: Prentice Hall Press, 2009.