Daniel Pickem, Andrew Melim, Jarius Tillman, Matheus Svolenski

Deliverables include the following:

- A 4-5 page PDF summary that describes your work, results and answers to the questions posed below. Thoroughness in analysis and answers in the reports are the primary component of your grade!
- A page in your summary describing what each group member did to participate in the project.
- A zip le that includes your code and results in an organized form.
- A README file that describes how to run the code.
- Print the summary and bring it to the following class

Problem 1 - Pre-Project: Towers of Hanoi

1. Explain the method by which each of the two planners nds a solution.
2. Which planner was fastest?
3. Explain why the winning planner might be more effective on this problem.

Problem 2 - Project Part I: Sokoban PDDL

1. Show successful plans from at least one planner on the three Sokoban problems in Figure 2 (1-3). The challenge problem is optional.
2. Compare the performance of two planners on this domain. Which one works better? Does this make sense, why?
3. Clearly PDDL was not intended for this sort of application. Discuss the challenges in expressing geometric constraints in semantic planning.
4. In many cases, geometric and dynamic planning are insufficient to describe a domain. Give an example of a problem that is best suited for semantic (classical) planning. Explain why a semantic representation would be desirable.

Problem 3 - Project Part II: Sokoban Planner

1. Give successful plans from your planner on the Sokoban problems in Figure 2 and any others.
2. Compare the performance of your planner to the PDDL planners you used in the previous problem. Which was faster? Why?
3. Prove that your planner was complete. Your instructor has a math background: a proof is a convincing argument. Make sure you address each aspect of completeness and why your planner satisfies it. Pictures are always welcome.
4. What methods did you use to speed up the planning? Give a short description of each method and explain why it did or didn't help on each relevant problem.

Problem 4 - Post-Project: Towers of Hanoi Revisited

1. Give successful plans from at least one planner with 6 and 10 disks.
2. Do you notice anything about the structure of the plans? Can you use this to increase the efficiency of planning for Towers of Hanoi? Explain.
3. In a paragraph or two, explain a general planning strategy that would take advantage of problem structure. Make sure your strategy applies to problems other than Towers of Hanoi. Would such a planner still be complete?

======= [12pt]article amsmath amssymb,amsmath graphics,graphicx,amssymb,verbatim,color epstopdf,amsxtra,epsfig, psfrag, subfig titlesec [toc,page]appendix

Daniel Pickem, Andrew Melim, Jarius Tillman, Matheus Svolenski

RIP - Robot Intelligence - Planning CS 4649/7649 Fall 2013 Assignment 1 04/18/2013

Deliverables include the following:

- A 4-5 page PDF summary that describes your work, results and answers to the questions posed below. Thoroughness in analysis and answers in the reports are the primary component of your grade!
- A page in your summary describing what each group member did to participate in the project.
- A zip le that includes your code and results in an organized form.
- A README file that describes how to run the code.
- Print the summary and bring it to the following class.

Problem 1 - Pre-Project: Towers of Hanoi

1. Explain the method by which each of the two planners nds a solution.
2. Which planner was fastest?
3. Explain why the winning planner might be more effective on this problem.

Problem 2 - Project Part I: Sokoban PDDL

1. **Show successful plans from at least one planner on the three Sokoban problems in Figure 2 (1-3). The challenge problem is optional.** The plans for problem 2.1 to 2.3 as well as the challenge problem are shown in the appendix.
2. **Compare the performance of two planners on this domain. Which one works better? Does this make sense, why?**
An overview of the numerical results is given in Table **??**. We compared two planners, namely Fast Forward and BlackBox.

    - BlackBox (uses GraphPlan and SAT - see [2])

Table 1: Performance Comparison of BlackBox and FF.

Problem Instance	**FF (default)**	**BlackBox (default)**	**BlackBox (walksat)**
Problem 2.1 [time]	0	0	0
Problem 2.1 [steps]	0	0	0
Problem 2.2 [time]	0	0	0
Problem 2.2 [steps]	0	0	0
Problem 2.3 [time]	0	0	0
Problem 2.3 [steps]	0	0	0
Problem Challenge [time]	0	0	0
Problem Challenge [steps]	0	0	0

- Fast Forward (see [3])

  those produced by members of the respective relaxed solution. FF employs a slightly more elaborated form of this heuristic, which we call helpful actions pruning. The simple architecture described so far already solves most of the available benchmarks extremely efficiently. Problematic cases are when there are dead ends — states from which the goals get unreachable — or goal orderings. In the presence of the latter phenomenon, like in the Blocksworld, the local search sometimes proceeds too greedily, and gets trapped. To overcome this, we have integrated the Goal Agenda algorithm (first proposed by Jana Koehler), as well as a simple goal ordering technique of our own, based on the relaxed solutions. In order to deal with dead end states, which can cause the search to fail entirely, we have chosen a simple safety-net solution: if local search fails, then we skip everything done so far and switch to a complete best-first algorithm that simply expands all search nodes by increasing order of goal distance evaluation.

3. **Clearly PDDL was not intended for this sort of application. Discuss the challenges in expressing geometric constraints in semantic planning.**

   - PDDL can't handle continuous worlds, the world has to be discretized.
   - There is no inherent notion of adjacency or neighborhood. Such relations have to be expressed explicitly. E.g. if squares are used, for each square PDDL requires 4 adjacency clauses.
   - In PDDL each square has to be given a label instead of just saying if the robot moves one step to the right, its x-coordinate increased by one. This leads to a large amount of labels and adjacency relationships in the problem description.
   - It needs to be explicitly stated whether a square is empty or contains a box or robot in order to detect an obstacle. Each action taken must ensure to update the 'empty' property of a square. This is a classical manifestation of the frame problem.
   - Even for small problem instances as the ones shown in the examples it was too tedious to write down the PDDL instantiation by hand. What could be described easily as a matrix required roughly 100 lines of PDDL code.

4. **In many cases, geometric and dynamic planning are insufficient to describe a domain. Give an example of a problem that is best suited for semantic (classical) planning. Explain why a semantic representation would be desirable.**

    - Clearly, geometric constraints can't be efficiently expressed in PDDL. Therefore, all problem types that have a physical interpretation and an embodiment in a physical and geometric world are not well suited for PDDL. Domains with geometric constraints that encode for example positions of objects or distances between objects (in continuous or discrete space) are best handled by geometric planners.
    - Problems that represent an abstract world, however, where neither states nor actions are tied to a geometric interpretation of the world are well suited for PDDL and classical semantic planning. In these types of problems dimensions/size of objects do not matter and actions are not tied to geometric constraints. In this case constraints define the abstract rules of the domain rather than a geometric setup.
    - A very classic problem in semantic planning is for example the 'change tire problem' (see [1]) that describes how to change a flat tire on a car. The *move* action moves objects from one abstract location to another (e.g. from the trunk to an axle). These locations hold no geometric information and are essentially just labels for objects in the world. The dimensions of objects do not matter either. Such a domain where the problem has been abstracted away from the geometric constraints are well-suited for a semantic description such as PDDL.

Problem 3 - Project Part II: Sokoban Planner

1. Give successful plans from your planner on the Sokoban problems in Figure 2 and any others.
2. Compare the performance of your planner to the PDDL planners you used in the previous problem. Which was faster? Why?
3. Prove that your planner was complete. Your instructor has a math background: a proof is a convincing argument. Make sure you address each aspect of completeness and why your planner satisfies it. Pictures are always welcome.
4. What methods did you use to speed up the planning? Give a short description of each method and explain why it did or didn't help on each relevant problem.

Problem 4 - Post-Project: Towers of Hanoi Revisited

1. Give successful plans from at least one planner with 6 and 10 disks.
2. Do you notice anything about the structure of the plans? Can you use this to increase the efficiency of planning for Towers of Hanoi? Explain.

3. In a paragraph or two, explain a general planning strategy that would take advantage of problem structure. Make sure your strategy applies to problems other than Towers of Hanoi. Would such a planner still be complete?

Summary

This section summarizes the contribution of each team member.

1. Matheus Svolenski
2. Jarius Tillman
3. Andrew Melim
4. Daniel Pickem
    - Implemented the class structure of the planner from problem 3.
    - Implemented the world class for the planner that describes and stores the environment and Sokoban map.
    - Implemented the A* algorithm.
    - Created PDDL description for problem 2.
    - Ran BlackBox and FF planners on the Towers of Hanoi problem 1 and problem 4.
    - Wrote Matlab tool to autogenerate Tower of Hanoi PDDL description.
    - Wrote Matlab tool to autogenerate Sokoban PDDL description.

References

[1] Russell, Stuart J. and Norvig, Peter, *Artificial Intelligence: A Modern Approach*, Pearson Education, 2010

[2] BlackBox Planner, *http://www.cs.rochester.edu/ kautz/satplan/blackbox/#super*

[3] Fast Forward Planner, *http://fai.cs.uni-saarland.de/hoffmann/ff.html*

Problem 2 - Project Part I: Sokoban PDDL - Successful Plans

- Problem 2.1:

  1. : MOVE R S-3-3 S-3-4 UP
  2. : MOVE R S-3-4 S-4-4 RIGHT
  3. : MOVE R S-4-4 S-5-4 RIGHT
  4. : MOVE R S-5-4 S-5-5 UP
  5. : PUSH R B1 S-5-5 S-4-5 S-3-5 LEFT
  6. : MOVE R S-4-5 S-4-4 DOWN
  7. : MOVE R S-4-4 S-3-4 LEFT
  8. : MOVE R S-3-4 S-2-4 LEFT
  9. : MOVE R S-2-4 S-2-5 UP
  10. : MOVE R S-2-5 S-2-6 UP
  11. : MOVE R S-2-6 S-3-6 RIGHT
  12. : PUSH R B1 S-3-6 S-3-5 S-3-4 DOWN
  13. : PUSH R B1 S-3-5 S-3-4 S-3-3 DOWN
  14. : PUSH R B1 S-3-4 S-3-3 S-3-2 DOWN

- Problem 2.2:

  1. : MOVE R S-2-3 S-2-2 DOWN
  2. : MOVE R S-2-2 S-3-2 RIGHT
  3. : MOVE R S-3-2 S-4-2 RIGHT
  4. : PUSH R B2 S-4-2 S-4-3 S-4-4 UP
  5. : MOVE R S-4-3 S-4-2 DOWN
  6. : MOVE R S-4-2 S-3-2 LEFT
  7. : MOVE R S-3-2 S-2-2 LEFT
  8. : MOVE R S-2-2 S-2-3 UP
  9. : PUSH R B1 S-2-3 S-3-3 S-4-3 RIGHT
  10. : MOVE R S-3-3 S-3-4 UP
  11. : PUSH R B2 S-3-4 S-4-4 S-5-4 RIGHT
  12. : MOVE R S-4-4 S-3-4 LEFT
  13. : MOVE R S-3-4 S-3-3 DOWN
  14. : MOVE R S-3-3 S-3-2 DOWN

15. : MOVE R S-3-2 S-4-2 RIGHT
16. : PUSH R B1 S-4-2 S-4-3 S-4-4 UP
17. : MOVE R S-4-3 S-5-3 RIGHT
18. : PUSH R B2 S-5-3 S-5-4 S-5-5 UP
19. : PUSH R B2 S-5-4 S-5-5 S-5-6 UP
20. : PUSH R B2 S-5-5 S-5-6 S-5-7 UP
21. : MOVE R S-5-6 S-5-5 DOWN
22. : MOVE R S-5-5 S-4-5 LEFT
23. : PUSH R B1 S-4-5 S-4-4 S-4-3 DOWN
24. : MOVE R S-4-4 S-5-4 RIGHT
25. : MOVE R S-5-4 S-5-3 DOWN
26. : PUSH R B1 S-5-3 S-4-3 S-3-3 LEFT
27. : MOVE R S-4-3 S-4-4 UP
28. : MOVE R S-4-4 S-3-4 LEFT
29. : PUSH R B1 S-3-4 S-3-3 S-3-2 DOWN
30. : MOVE R S-3-3 S-4-3 RIGHT
31. : MOVE R S-4-3 S-4-2 DOWN
32. : PUSH R B1 S-4-2 S-3-2 S-2-2 LEFT

- Problem 2.3 0: MOVE R S-3-5 S-4-5 RIGHT 1: MOVE R S-4-5 S-4-6 UP 2: MOVE R S-4-6 S-4-7 UP 3: MOVE R S-4-7 S-5-7 RIGHT 4: MOVE R S-5-7 S-6-7 RIGHT 5: MOVE R S-6-7 S-6-6 DOWN 6: MOVE R S-6-6 S-6-5 DOWN 7: PUSH R B1 S-6-5 S-5-5 S-4-5 LEFT 8: PUSH R B1 S-5-5 S-4-5 S-3-5 LEFT 9: MOVE R S-4-5 S-5-5 RIGHT 10: MOVE R S-5-5 S-6-5 RIGHT 11: MOVE R S-6-5 S-6-6 UP 12: MOVE R S-6-6 S-6-7 UP 13: MOVE R S-6-7 S-7-7 RIGHT 14: MOVE R S-7-7 S-8-7 RIGHT 15: MOVE R S-8-7 S-8-6 DOWN 16: MOVE R S-8-6 S-8-5 DOWN 17: MOVE R S-8-5 S-8-4 DOWN 18: MOVE R S-8-4 S-8-3 DOWN 19: MOVE R S-8-3 S-8-2 DOWN 20: MOVE R S-8-2 S-9-2 RIGHT 21: MOVE R S-9-2 S-10-2 RIGHT 22: MOVE R S-10-2 S-10-3 UP 23: MOVE R S-10-3 S-10-4 UP 24: MOVE R S-10-4 S-10-5 UP 25: PUSH R B3 S-10-5 S-9-5 S-8-5 LEFT 26: MOVE R S-9-5 S-10-5 RIGHT 27: MOVE R S-10-5 S-10-4 DOWN 28: MOVE R S-10-4 S-10-3 DOWN 29: MOVE R S-10-3 S-10-2 DOWN 30: MOVE R S-10-2 S-9-2 LEFT 31: MOVE R S-9-2 S-8-2 LEFT 32: MOVE R S-8-2 S-8-3 UP 33: MOVE R S-8-3 S-8-4 UP 34: PUSH R B3 S-8-4 S-8-5 S-8-6 UP 35: PUSH R B2 S-8-5 S-7-5 S-6-5 LEFT 36: PUSH R B2 S-7-5 S-6-5 S-5-5 LEFT 37: MOVE R S-6-5 S-6-6 UP 38: MOVE R S-6-6 S-6-7 UP 39: MOVE R S-6-7 S-5-7 LEFT 40: MOVE R S-5-7 S-4-7 LEFT 41: MOVE R S-4-7 S-4-6 DOWN 42: MOVE R S-4-6 S-4-5 DOWN 43: PUSH R B2 S-4-5 S-5-5 S-6-5 RIGHT 44: PUSH R B2 S-5-5 S-6-5 S-7-5 RIGHT 45: MOVE R S-6-5 S-5-5 LEFT 46: MOVE R S-5-5 S-4-5 LEFT 47: MOVE R S-4-5 S-4-6 UP 48: MOVE R S-4-6 S-4-7 UP 49: MOVE R S-4-7 S-3-7 LEFT 50: MOVE R S-3-7 S-2-7 LEFT 51: MOVE

R S-2-7 S-2-6 DOWN 52: MOVE R S-2-6 S-2-5 DOWN 53: PUSH R B1 S-2-5 S-3-5 S-4-5 RIGHT 54: PUSH R B1 S-3-5 S-4-5 S-5-5 RIGHT 55: MOVE R S-4-5 S-4-6 UP 56: MOVE R S-4-6 S-4-7 UP 57: MOVE R S-4-7 S-5-7 RIGHT 58: MOVE R S-5-7 S-6-7 RIGHT 59: MOVE R S-6-7 S-6-6 DOWN 60: MOVE R S-6-6 S-6-5 DOWN 61: PUSH R B1 S-6-5 S-5-5 S-4-5 LEFT 62: MOVE R S-5-5 S-6-5 RIGHT 63: MOVE R S-6-5 S-6-6 UP 64: MOVE R S-6-6 S-6-7 UP 65: MOVE R S-6-7 S-7-7 RIGHT 66: MOVE R S-7-7 S-8-7 RIGHT 67: PUSH R B3 S-8-7 S-8-6 S-8-5 DOWN 68: PUSH R B3 S-8-6 S-8-5 S-8-4 DOWN 69: PUSH R B3 S-8-5 S-8-4 S-8-3 DOWN 70: PUSH R B3 S-8-4 S-8-3 S-8-2 DOWN 71: MOVE R S-8-3 S-8-4 UP 72: MOVE R S-8-4 S-8-5 UP 73: MOVE R S-8-5 S-8-6 UP 74: MOVE R S-8-6 S-8-7 UP 75: MOVE R S-8-7 S-7-7 LEFT 76: MOVE R S-7-7 S-6-7 LEFT 77: MOVE R S-6-7 S-6-6 DOWN 78: MOVE R S-6-6 S-6-5 DOWN 79: PUSH R B2 S-6-5 S-7-5 S-8-5 RIGHT 80: MOVE R S-7-5 S-6-5 LEFT 81: MOVE R S-6-5 S-6-6 UP 82: MOVE R S-6-6 S-6-7 UP 83: MOVE R S-6-7 S-7-7 RIGHT 84: MOVE R S-7-7 S-8-7 RIGHT 85: MOVE R S-8-7 S-8-6 DOWN 86: PUSH R B2 S-8-6 S-8-5 S-8-4 DOWN 87: PUSH R B2 S-8-5 S-8-4 S-8-3 DOWN 88: MOVE R S-8-4 S-8-5 UP 89: MOVE R S-8-5 S-7-5 LEFT 90: MOVE R S-7-5 S-6-5 LEFT 91: MOVE R S-6-5 S-6-6 UP 92: MOVE R S-6-6 S-6-7 UP 93: MOVE R S-6-7 S-5-7 LEFT 94: MOVE R S-5-7 S-4-7 LEFT 95: MOVE R S-4-7 S-3-7 LEFT 96: MOVE R S-3-7 S-2-7 LEFT 97: MOVE R S-2-7 S-2-6 DOWN 98: MOVE R S-2-6 S-2-5 DOWN 99: MOVE R S-2-5 S-3-5 RIGHT 100: PUSH R B1 S-3-5 S-4-5 S-5-5 RIGHT 101: PUSH R B1 S-4-5 S-5-5 S-6-5 RIGHT 102: PUSH R B1 S-5-5 S-6-5 S-7-5 RIGHT 103: PUSH R B1 S-6-5 S-7-5 S-8-5 RIGHT 104: MOVE R S-7-5 S-6-5 LEFT 105: MOVE R S-6-5 S-6-6 UP 106: MOVE R S-6-6 S-6-7 UP 107: MOVE R S-6-7 S-7-7 RIGHT 108: MOVE R S-7-7 S-8-7 RIGHT 109: MOVE R S-8-7 S-8-6 DOWN 110: PUSH R B1 S-8-6 S-8-5 S-8-4 DOWN

- Problem 2.4 Challenge 0: MOVE R S-6-4 S-6-5 UP 1: MOVE R S-6-5 S-5-5 LEFT 2: MOVE R S-5-5 S-4-5 LEFT 3: MOVE R S-4-5 S-3-5 LEFT 4: MOVE R S-3-5 S-2-5 LEFT 5: MOVE R S-2-5 S-2-4 DOWN 6: PUSH R B1 S-2-4 S-3-4 S-4-4 RIGHT 7: PUSH R B1 S-3-4 S-4-4 S-5-4 RIGHT 8: MOVE R S-4-4 S-4-5 UP 9: MOVE R S-4-5 S-4-6 UP 10: MOVE R S-4-6 S-4-7 UP 11: MOVE R S-4-7 S-3-7 LEFT 12: PUSH R B2 S-3-7 S-3-6 S-3-5 DOWN 13: MOVE R S-3-6 S-4-6 RIGHT 14: MOVE R S-4-6 S-4-5 DOWN 15: MOVE R S-4-5 S-4-4 DOWN 16: MOVE R S-4-4 S-3-4 LEFT 17: MOVE R S-3-4 S-2-4 LEFT 18: MOVE R S-2-4 S-2-5 UP 19: PUSH R B2 S-2-5 S-3-5 S-4-5 RIGHT 20: MOVE R S-3-5 S-3-6 UP 21: MOVE R S-3-6 S-4-6 RIGHT 22: MOVE R S-4-6 S-5-6 RIGHT 23: MOVE R S-5-6 S-5-5 DOWN 24: MOVE R S-5-5 S-6-5 RIGHT 25: MOVE R S-6-5 S-6-4 DOWN 26: PUSH R B1 S-6-4 S-5-4 S-4-4 LEFT 27: PUSH R B1 S-5-4 S-4-4 S-3-4 LEFT 28: MOVE R S-4-4 S-5-4 RIGHT 29: MOVE R S-5-4 S-5-5 UP 30: MOVE R S-5-5 S-5-6 UP 31: MOVE R S-5-6 S-4-6 LEFT 32: MOVE R S-4-6 S-3-6 LEFT 33: MOVE R S-3-6 S-3-5 DOWN 34: PUSH R B2 S-3-5 S-4-5 S-5-5 RIGHT 35: MOVE R S-4-5 S-4-4 DOWN 36: MOVE R S-4-4 S-5-4 RIGHT 37: MOVE R S-5-4 S-5-3 DOWN 38: MOVE R S-5-3 S-5-2 DOWN 39: MOVE R S-5-2 S-4-2 LEFT 40: PUSH R B3 S-4-2 S-4-3 S-4-4 UP 41: PUSH R B3 S-4-3 S-4-4 S-4-5 UP 42: PUSH R B3 S-4-4 S-4-5 S-4-6 UP 43: MOVE R S-4-5 S-3-5 LEFT 44: MOVE R S-3-5 S-2-5 LEFT 45: MOVE R S-2-5 S-2-4 DOWN 46: PUSH R B1 S-2-4 S-3-4 S-4-4 RIGHT 47: MOVE R S-3-4 S-3-5 UP 48: MOVE R S-3-5 S-4-5 RIGHT 49: PUSH R B1 S-4-5 S-4-4 S-4-3 DOWN 50: MOVE R S-4-4 S-5-4 RIGHT 51: MOVE R S-5-4 S-6-4 RIGHT 52: MOVE R S-6-4 S-6-5 UP

53: PUSH R B2 S-6-5 S-5-5 S-4-5 LEFT 54: PUSH R B2 S-5-5 S-4-5 S-3-5 LEFT 55: MOVE R S-4-5 S-5-5 RIGHT 56: MOVE R S-5-5 S-5-6 UP 57: PUSH R B3 S-5-6 S-4-6 S-3-6 LEFT 58: MOVE R S-4-6 S-4-5 DOWN 59: MOVE R S-4-5 S-4-4 DOWN 60: MOVE R S-4-4 S-3-4 LEFT 61: MOVE R S-3-4 S-2-4 LEFT 62: MOVE R S-2-4 S-2-5 UP 63: PUSH R B2 S-2-5 S-3-5 S-4-5 RIGHT 64: PUSH R B2 S-3-5 S-4-5 S-5-5 RIGHT 65: MOVE R S-4-5 S-4-6 UP 66: MOVE R S-4-6 S-4-7 UP 67: MOVE R S-4-7 S-3-7 LEFT 68: PUSH R B3 S-3-7 S-3-6 S-3-5 DOWN 69: MOVE R S-3-6 S-4-6 RIGHT 70: MOVE R S-4-6 S-4-5 DOWN 71: MOVE R S-4-5 S-4-4 DOWN 72: MOVE R S-4-4 S-5-4 RIGHT 73: MOVE R S-5-4 S-5-3 DOWN 74: MOVE R S-5-3 S-5-2 DOWN 75: MOVE R S-5-2 S-4-2 LEFT 76: PUSH R B1 S-4-2 S-4-3 S-4-4 UP 77: PUSH R B1 S-4-3 S-4-4 S-4-5 UP