

Midterm

October 11, 2005

Due: 11:59 pm — October 24, 2005

Create a .tar file containing your answers to all the questions and submit to “Midterm” in the homework submission website..

There are 4 questions and a programming component. The purpose of this midterm is to remind you about the topics that we have gone through.

1. (First order logic)

(a) Representing the following statements in first order logic:

- Horses, cows, and pigs are mammal.
- An offspring of a horse is a horse.
- Bluebeard is a horse.
- Bluebeard is Charlie’s parent.
- Offspring and parent are inverse relation.
- Every mammal has a parent.

(b) Use the resolution procedure, show that Charlie is a horse. Provide the step-by-step computation that leads to the answer.

2. (CSP)

(a) Formulate the n-queens problem as a constraint satisfaction problem.

(b) Solve the 4-queens problem manually following the backtracking procedure. You can use AC-3 to prune the domains.

3. (Logic Programming)

Let Π be the following logic program

$$\begin{array}{ll} p(a) & \leftarrow \\ p(f(X)) & \leftarrow g(X) \\ g(Y) & \leftarrow p(Y) \end{array}$$

Compute $lfp(T_{\Pi})$.

4. (Answer Set Programming)

Select one of the following two problems and solve it using answer set programming, i.e., develop a logic program whose answer sets correspond one-to-one to solutions of the problem. Prove the correctness of your program.

- *Problem 1:* A king wants to invite his knights to a round table dinner. Unfortunately, not every knight is friend of everyone. And, none wants to sit next to those who are not his friends. Being considered, the king asks everybody to write down the name of his friends and also the names of his “unwanted” friends. He is looking for a sit ordering so that no one has to sit next to his unwanted friend and friends sit next to each other if possible. Can you help the king to find such an ordering?
- *Problem 2:* Given a collection C of set of integers from 1 to n . Is there a way to divide the set $\{1, \dots, n\}$ into two sets S_1 and S_2 such that $S_1 \cup S_2 = \{1, \dots, n\}$ and $S_1 \cap S_2 = \emptyset$ and every set in C is entirely contained in either S_1 or S_2 . For example, for $n = 4$, $C = \{\{1\}, \{2, 3\}, \{4\}\}$, $S_1 = \{1, 4\}$, and $S_2 = \{2, 3\}$, the answer is yes; however, the answer is no if $S_1 = \{1, 4, 2\}$, and $S_2 = \{3\}$.

5. Programming Component

In this component, you are asked to implement a planner. The planner should take a planning problem as input and return a solution (if one exists) or no solution (if none exists). On the class website, you will find the following:

- An example of a planner (planner.gz)
- A .tar file (domains.tar.gz) that contains some planning problems that you should use in testing your planner.
- A packet contains the source code of the parser (parser.tar.gz). The parser takes as input a planning problem written in the language \mathcal{AL} (see the .al files) and create a list of actions and the initial state (as a set).

Requirement:

- Implement the best first search algorithm using the ‘delete-list’ heuristic. As you have implemented the search algorithm before it should not be difficult. You will need to adapt your code so it works with the new data structure. Before doing so, you need to implement the heuristic function and the function for computing the next state.
- Make sure that your program yields reasonable performance. It should be able to solve at least 50% of the problems in the domains.tar file within 10 minutes.
- Record the problems that your planner cannot solve within 10 minutes.

Note 1: Tu Phan (tphan@cs.nmsu.edu) has provided me the code for the parser and the planner. He understood the code very well and would be interested in bugs if you have some to report.

Note 2: For bonus point, you can implement the progression/regression planner as described in Bonet and Geffner’s paper. Note that if you do so, you should use their heuristic function.