CSE 573 Problem Set 2

Answers due 10/28/08 8am

Preferably, turn in by email to both **daipeng@cs** & **weld@cs**

Please work on this problem set individually. If any problem doesn't contain enough information for you to answer it, feel free to make any assumptions necessary to find the answer, but state the assumptions clearly. You will be graded on your choice of assumptions as well as the clarity of your written answers.

We'll accept late problem sets under the following conditions:
Up to 1 hour late - no penalty
Up to one day late - 25% penalty
Up to two day late - 50% penalty
More than 2 days - no credit (please plan ahead!)

1. (15 points) R&N problem 7.4 parts, a, b and c.
2. (15 points) R&N problem 7.8 parts a, b, c and e.
3. (5 points) R&N problem 7.16
4. (5 points) R&N problem 8.2
5. (5 points) R&N problem 8.7 using Nationality(p, c) to say that p is from country c and the constant G to denote Germany.
6. (5 points) R&N problem 8.8
7. (5 points) R&N problem 8.13
8. (5 points) R&N problem 9.4
9. (30 points) Using the programming language of your choice, implement a function which performs unification. You may accept input in any convenient manner and use any syntax for variables and formulae. Hand in the code for your algorithm and examples showing its correctness, including the ones presented in class (slides of 10/14) as an explanation of the function, plus at least two others.
10. In this problem, you are asked to execute SatPlan (a modern planning algorithm, developed by Henry Kautz and colleagues, which compiles problems to SAT) on a past International Planning Competition (IPC) domain.
    a) First, download the executable of SatPlan from http://www.cs.washington.edu/education/courses/cse573/08au/problems/ps2/SatPlan2006_LinuxBin.tgz Before you run SatPlan, make sure you read the readme file carefully (available at http://www.cs.washington.edu/education/courses/cse573/08au/problems/ps2/README.txt). Please ignore the installation and "how to get started" section, since you've already got the executables.
    b) Next, download the "Storage" domain file and problem files from http://www.cs.washington.edu/education/courses/cse573/08au/problems/ps2/storage.tar . You might want to read the description of this domain from the IPC5 website http://zeus.ing.unibs.it/ipc-5/domain-descriptions/storage.txt
    c) The domains are implemented in PDDL (http://zeus.ing.unibs.it/ipc-5/pddl.html), a "industrial grade" version of the STRIPS language with which we defined actions in

class – you know, preconditions and postconditions, etc. You may wish to review the PDDL language now, in order to understand the domain encoding. You will *definitely* want to read about PDDL before trying the next problem.

d) Use the default parameters to solve the first problem (FYI, if you have put the untared **storage** folder exactly inside the untared **SatPlan2006_LinuxBin**, then the exact command input should be **"./satplan -domain storage/domain.pddl -problem storage/p01.pddl**"). This should produce a lot of detailed output, but don't get frightened - just ignore most of it for the moment. What you really care about is whether the problem is solved. If the problem is solved optimally, there should be a line says "***SAT!***". Layer tells you how many time steps are used to solve problem (multiple compatible actions may be scheduled for each level). Right below that line should be some timing information.

e) Now you should be able to solve the small problems (up to 9) instantly. If you want to see how SatPlan performs on some larger problems in this domain, try running problem 10. You might also notice that SatPlan fails to solve the large problems (e.g. 20).

f) (10 points) Run problem 2. In how many time steps this problem can be solved? (check the MakeSpan info in the solution file) Write down (in English) which actions are executed in each time step. (To do this, you need to comprehend the domain file and the problem file first, and join them with the solution file)

g) (5 points) Could one switch the actions in time 0 and time 1, (i.e., the order of the first two actions in the solution) and still have a working plan? If not, what pre-condition(s) is/are violated if you perform the second action first?

11. (25 points) Use the PDDL language to encode a domain and problem of your choice (mobile robots, getting a PhD, getting lunch, traveling to another city for a conference, dating, whatever) so a planner can solve it. Then use SatPlan to find a solution plan. Turn in your PDDL definitions of your domain, and at least one problem, the solution file for the first problem, and a qualitative description of the resulting plan (e.g., length number of actions, etc). Your grade will depend on the complexity of the domain and problem specifications with extra credit available. **Advice:** Start simply and test for solution plans frequently. Add actions incrementally. And likewise increase the complexity of preconditions and effects with frequent testing, since syntax errors may be hard to spot.