Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2016-10-20
Sal (1)	TER3
Tid	14-18
Kurskod	TDDD48
Provkod	TEN1
Kursnamn/benämning **Provnamn/benämning**	Automatisk planering Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	4
Jour/Kursansvarig Ange vem som besöker salen	Jonas Kvarnström
Telefon under skrivtiden	0704-737579
Besöker salen ca klockan	ja 15-16
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, ankn. 2362, anna.grabska.eklund@liu.se
Tillåtna hjälpmedel	inga
Övrigt	
Antal exemplar i påsen	

TDDD48 Automated Planning 2016-10-20

Important Instructions: Read before you begin!

- Though the questions are in English, feel free to **answer in Swedish** if you prefer! Det går bra att skriva på svenska!
- *Clear and comprehensible* explanations and motivations are always required. This does not necessarily mean that each answer should be a long essay. What is important is that all the relevant facts are present and clearly explained.
- Make an effort to write explanations that can be understood by someone who *does not already know* the correct answer. If your answer explains a topic well enough that a fellow student could *learn* something new from the answer and/or apply it in practice, then you have probably succeeded. Conversely, if we ask how HTNs work, saying that "there are methods and operators and you construct a tree using patterns and there can be constraints too, and there's no goal" does not provide much in terms of *useful* information, and certainly will not give you a full score.
- To see if you have succeeded, turn the tables and **try to misunderstand.** Specifically, after you write an answer, go back and see if you can creatively misinterpret the answer. If so, *clarify, clarify, clarify*!
- Concrete examples or counterexamples may be useful as part of a motivation. If so, please make sure you include all relevant information about the example you have chosen to use. What is relevant naturally depends on how you use the example.
- When asked to provide examples or illustrate something through a planning problem instance, save time by *keeping the example as small as possible.*

1 General Concepts in Automated Planning

a) Explain the difference between *domain-independent* and *domain-specific* planning. Also, for each of these two types of planning, describe one advantage it has over the other. **(2 points)**

b) Let $P_1 = (O, s_0, g_1)$ and $P_2 = (O, s_0, g_2)$ be two classical planning problems that share the same operators and initial state. Let $\pi_1 = [a_1, \ldots, a_n]$ be a solution of length n for P_1, and let $\pi_2 = [b_1, \ldots, b_m]$ be a solution of length m for P_2.

   Suppose that the concatenated action sequence $\pi_1 \cdot \pi_2 = [a_1, \ldots, a_n, b_1, \ldots, b_m]$ happens to be *executable* starting at state s_0. Can we then be certain that it is also a *solution* for P_2 (achieves the goals of P_2)? Show clearly that this *is* or *is not* the case. If you want to provide an example or counterexample, try to keep it short and simple. **(2 points)**

c) **Relaxation.** Let $P = \langle O, s_0, g \rangle$ be an arbitrary classical planning problem. When is $P' = \langle O', s'_0, g' \rangle$ a *relaxed* version of P? That is, what criterion or criteria must be satisfied for P' to be a relaxation of P?

   Note that we are discussing relaxation in general, as opposed to some specific type of relaxation. Note also that you should describe the *exact* requirements for relaxation. That is, your definition should be *sufficient and necessary*, rather than covering a special case. **(2 points)**

2 Partial-Order Planning

Partial-order causal link (POCL) planning differs from forward state space search by generating partially ordered plans. This requires a different search space as well as a different search method.

Name one important strength that these differences give POCL planning over forward state space planning, as well as one important strength that the differences give forward state space planning over POCL planning, when it comes to finding a feasible plan in a reasonable amount of time.

Make sure that you explain why these strengths exist and why they are not shared by the other planning paradigm. For example, techniques such as lifted planning can be applied to both of the planning methods mentioned above and are therefore not strengths of one method *over* the other. Features that are inherent in one method and not in the other, or techniques that can be applied to one method but not to the other, are of greater interest. **(3 points)**

3 Search Guidance and Heuristics

Given the size of a typical search space, a planner almost always needs some form of search guidance as opposed to doing blind search. Often such guidance takes the shape of a *heuristic function* that evaluates the "quality" of a certain search node, corresponding to one specific choice that can be made at a particular point in the search space.

a) Heuristic functions can yield *plateaus* in the search space. What is a plateau? Visualize one using part of a search space: Show a set of search nodes, a number of possible transitions between those nodes, and the type of heuristic values that characterize a plateau. Show which nodes are part of the plateau and explain in words *why* they form a plateau. **(1 point)**

b) Some search algorithms used in automated planning originate in general *optimization* problems, where plateaus can in some cases be handled simply by terminating search as soon as a plateau is reached. Explain why this approach is generally not useful when these search algorithms are instead applied to classical planning problems. **(1 point)**

c) Is the h_{add} heuristic, also called h_0, admissible?

If it is, motivate why the calculations will always lead to an admissible heuristic value. Write a clear motivation that shows why admissibility can never be violated by the calculations.

If it is not, demonstrate using a small counterexample that includes a current state, a goal to be achieved, and a set of actions where the heuristic function can yield an inadmissible heuristic value. Remember to include sufficient information about the actions so that your claim can be proven. **(2 points)**

d) Many heuristics work by solving *subproblems* of the actual problem being solved, and then combining the costs of these subproblems in some way. For example, a single subproblem for the h_2 heuristic is constructed by selecting two literals from the "real" goal, and then (under)estimating the cost of achieving only these two literals. The heuristic then solves one such subproblem for *every* pair of goal literals, and returns the maximum of all calculated costs.

What is the corresponding way in which subproblems are constructed for a *pattern database heuristic*? Describe the general steps and ideas involved. Assume that the problem is already in state-variable representation as opposed to classical representation, so that an initial conversion phase (as described during the lectures) is not necessary. **(2 points)**

4 Markov Decision Processes

a) The states and transitions involved in a classical planning problem can be described using a specific type of state transition system. This is also the case for the states and transitions involved in an MDP, a (fully observable) Markov Decision Process.

However, there is a key difference in expressivity between classical planning problems and MDP problems, which is also reflected in the definition of their respective state transition systems. What is this difference (what can you model in an MDP but not in classical planning)?

Also, how is the increased expressivity represented in the state transition system of an MDP? (Here, "state transition system" means the tuple containing a set of states, a set of actions, and so on.) **(1 point)**

b) Markov Decision Processes are characterized by the *Markov property*. What is the Markov property? **(1 point)**

c) Markov Decision Processes often use a *discount factor*. Which property of a policy is calculated using the discount factor, and how is the discount factor used in the calculations? You can express the answer as a formula or in words, as long as the description clearly shows that you understand how the discount factor is used. Also explain why the use of a discount factor is not only a mathematical trick but in many cases results in a better model of our preferences/goals/desires than if the discount factor had not been used. **(2 points)**

d) *Policy iteration* and *value iteration* are two standard iterative methods for finding suitable policies for an MDP. What is the main difference between these methods (in terms of what they *do* in each iteration)? What consequences can this difference have in terms of performance and convergence towards an optimal policy for an MDP? **(2 points)**