Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2014-08-21
Sal (1) Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och ringa in vilken sal som avses	TER3
Tid	8-12
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 kl.	Ja
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund, ankn. 2362, anna.grabska.eklund@liu.se
Tillåtna hjälpmedel	inga
Övrigt	
Vilken typ av papper ska användas, rutigt eller linjerat	Valfritt
Antal exemplar i påsen	

Exam: TDDD48 Automated Planning 2014-08-21

Important Notes

Read the following before you begin!

- Though the questions are in English, feel free to **answer in Swedish** if you prefer!
- *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. Write an explanation that can be understood and applied by someone who does not already know the answer!
- 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 Fundamental Concepts

a) Consider the standard Blocks World domain, with one type (*block*) and five predicates: *ontable(block)*, *clear(block)*, *on(block,block)*, *holding(block)*, and *handempty*.

If a given planning problem instance for this domain contains n blocks, *exactly* how many states are there in the state space?

Assume that in the initial state s_0, all blocks are clear and on the table, no block is on another block, no block is being held, and the hand is empty. This is one example of a physically achievable state. Give an *estimate* of how many other physically achievable states might be reachable from this initial state. The number does not have to be exact, but you must take into account at least some of the dependencies between propositions that cause certain states to be unreachable. For example, no state where the hand is empty while you are holding a block will be reachable. **(2 points)**

b) *Lifted* planning is a general technique applicable to a variety of search spaces and planning algorithms. During the lectures, we specifically discussed *lifted partial-order planning.*

Explain what lifted planning is in general, and give a concrete example of why it can be useful when generating partial-order plans. In this example you should show a small lifted partial-order plan structure (or a reasonable fragment of one), contrast it against a non-lifted alternative, and explain what is better about the lifted version. **(2 points)**

2 Search Guidance

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) Many heuristic functions are based on *relaxations* of planning problems.

Let $P = (\Sigma, s_0, S_g)$ and $P' = (\Sigma', s_0', S_g')$ be two classical planning problems. When is P' a relaxation of P? We are looking for the actual definition of relaxation, not for specific ways of constructing a relaxation and not for any of the many secondary consequences. In other words: "P' is a relaxation of P if and only if ...". **(2 points)**

b) Let $P = (\Sigma, s_0, S_g)$ be a classical planning problem and $P' = (\Sigma', s_0', S_g')$ be a relaxation of that problem. What can you say about the relation between the cost of an arbitrary solution to P and an arbitrary solution to P'? Why? **(2 points)**

c) Explain one way in which an automated planner can "mechanically" relax a planning problem.

Note: The answer could be exemplified by showing a problem instance and its relaxation, but this cannot be the entire answer. You must provide a sufficiently general description of how the relaxation should be done, a "procedure" that could be applied to an arbitrary problem instance. **(2 points)**

3 Domain-Configurable Planning

a) Some domain-configurable planners allow the specification of simple *state constraints*.

What *is* a state constraint? In other words, what additional information can be expressed using a state constraint compared to what can be expressed in an "ordinary" classical planning problem?

Give a concrete example of a state constraint that could help a domain-configurable planner in a domain of your choice. You do not have to use any specific syntax; a comprehensible description of the constraint is sufficient.

Explain how a forward-chaining state space planner can concretely make use of such a constraint: When during the search process does it use the constraint, and how can this affect the search process? **(3 points)**

4 Path Planning

a) When generating a path for a robot, it is important to know whether the robot is *holonomic.*

What is the difference between holonomic and non-holonomic robots? You do not need to explain which one is which, as long as you clearly show what the *difference* is.

Give an example of a holonomic robot and one example of a non-holonomic robot. Again, labeling them correctly is not part of the exam. **(2 points)**

b) There are many methods for geometrically placing possible nodes and edges in a roadmap graph. You should pick *either* visibility graphs *or* Voronoi diagrams, and then:

- Draw a rectangular area with several (3–5) obstacles
- Show where the method you chose would place edges / possible paths through your area
- Explain why the paths are placed as they are – in other words, what the method actually does.

Again, you should pick *one* of the alternatives. If you try to hedge your answer by picking both, then both answers will have to be correct. **(2 points)**

c) In path planning, what is the *work space* and what is the *configuration space*? Explain clearly, preferably using small examples. **(2 points)**