Automated Planning
Example Exam

Jendrik Seipp
Linköping University
Department of Computer and Information Science

Name: ______________________________

- You may prepare and use **one sheet of A4 paper with notes** (using both sides). Other aids such as lecture slides, books, or calculators are not allowed. All electronic devices (such as mobile phones) must be switched off. You are allowed to use pen and paper for drafting your solutions.
- In case you develop several partial solutions for a question, please only keep the one that should be marked.
- Entering formulas and special symbols in WiseFlow is tricky. Therefore, please use LaTeX notation or understandable abbreviations such as the following examples:

Meaning	Abbreviation
$\mapsto$ (`\mapsto`)	`|->`
$\to$ (`\to`)	`-->`
$\triangleright$ (for conditional effects)	`->`
$s[\![o]\!]$	`s[o]`
$\wedge$	`and` or `^` (caret/circumflex)
$\vee$	`or` or `V` (uppercase)
$\neg$	`not` or ~
$\top$ and $\bot$	`T` and `F`
$\langle$ and $\rangle$	`<` and `>`
$\emptyset$	`emptyset`
o_1, I_1, etc.	`o1`, `I1`, etc.
h^{FF}, h^{add}, h^{+}, etc.	`hFF` and `hadd`, `h+` etc.
$\leq$ and $\geq$	`<=` and `>=`

Multiple-Choice Questions (5×2 marks)

Note: in every block, any number of answers (0–4) may be correct. For each block you get 2 marks if you select all and only the correct answers, and 0 marks otherwise.
Note that $|x|$ denotes the length or number of elements of x, and $\|x\|$ denotes the representation size of x.

(a) Which of the following statements are true for all propositional planning tasks $\Pi = \langle V, I, O, \gamma\rangle$?

- ☐ There are algorithms that solve Π in polynomial time in $\|\Pi\|$.
- ☐ There are algorithms that solve Π in polynomial time in the number of states of Π.
- ☐ Π has $2^{|V|}$ states.
- ☐ γ describes a single state of Π.

(b) Which of the following statements about planning formalisms, normal forms and mutexes are correct?

- ☐ The main difference between STRIPS and SAS^+ tasks is that state variables in SAS^+ tasks can take on more than two different values.
- ☐ A propositional planning task Π can be converted to positive normal with a polynomial increase in size.
- ☐ Mutexes are useful for computing the delete relaxation of a planning task.
- ☐ If the goal of a STRIPS planning task contains two propositions that are mutex, the task is unsolvable.

(c) Which of the following statements about delete relaxation are true?

- ☐ If the delete relaxation of a task Π in positive normal form is solvable, then Π is solvable.
- ☐ If the propositional task Π in positive normal form is solvable, then the corresponding delete-relaxed task is solvable.
- ☐ The h^{add} heuristic is often preferred over h^{FF} because it is admissible.
- ☐ The greedy algorithm for relaxed planning tasks results in an admissible heuristic.

(d) Which of the following statements about abstraction heuristics are true?

- ☐ Every merge-and-shrink heuristic can also be computed with a pattern database (PDB).
- ☐ Every PDB heuristic can also be computed within the merge-and-shrink framework.
- ☐ If we only merge and never shrink and never reduce labels, we obtain a transition system that is isomorphic to the transition system of the input task.
- ☐ Label reduction reduces the time and memory requirement for merge and shrink steps.

(e) Let $\mathcal{C} = \bigcup_{i=1}^{n} C_i$ and $\mathcal{C}'$ be sets of operator counting constraints. Which of the following statements about operator counting heuristics are true?

- ☐ Every feasible solution for $h^{LP}_{\mathcal{C}\cup\mathcal{C}'}$ is a feasible solution for $h^{LP}_{\mathcal{C}}$.
- ☐ $h^{LP}_{\mathcal{C}}$ is the optimal non-negative cost partitioning over the heuristics $h^{LP}_{C_i}$.
- ☐ $h^{LP}_{\mathcal{C}\cup\mathcal{C}'} \leq h^{LP}_{\mathcal{C}} + h^{LP}_{\mathcal{C}'}$
- ☐ There is an optimal plan that uses an operator o at least x times if $\text{Count}_o = x$ in a minimal solution for $h^{LP}_{\mathcal{C}}$.

Question 1 (Foundations, 1+1+1+2+2+3 marks)

Consider the planning task $\Pi = \langle \{u, v, w\}, \{u \mapsto \mathbf{F}, v \mapsto \mathbf{F}, w \mapsto \mathbf{F}\}, \{o_1, o_2, o_3\}, \neg u \land v \rangle$ with

- $o_1 = \langle \top, (u \rhd \neg u) \land (\neg u \rhd u), 1 \rangle$
- $o_2 = \langle u, v \land w, 1 \rangle$
- $o_3 = \langle \top, w, 1 \rangle$

(a) How many states does Π have? (Note: we do not ask for *reachable* states.)

(b) How many goal states does Π have?

(c) How many states have an outgoing transition labeled with o_2?

(d) Specify a plan that solves Π and state its cost.

(e) Give an example of a state of Π that is unreachable.

(f) Is $o = \langle a \lor \neg b, \neg c \land (c \rhd (\neg d \rhd (a \land b))) \rangle$ applicable in state $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}, d \mapsto \mathbf{F}\}$? Justify your answer. If yes, what is the resulting state $s[\![o]\!]$?

Unless explicitly specified, you do not need to justify your answers or give intermediate results for this question.

Question 2 (Planning algorithms, 3+2+2+3 marks)

(a) What is the STRIPS regression $sregr(\varphi, o)$ for $\varphi = (a \land b)$ and $o = \langle c, a \land \neg c \rangle$? Simplify the result as much as possible.

(b) In a regression search, when we expand a search state (= state formula) φ and regress it through an operator o, we can prune (discard) the resulting new search state if $regr(\varphi, o) \models \varphi$. Why can we do this (1–2 sentences)?

(c) Choose any algorithm for classical planning considered in the lecture and state whether its worst-case *runtime* and whether its worst-case *memory usage* are polynomial in the input size, i.e., the length of the planning task representation. Justify your answers (1–2 sentences).

(d) Name two heuristics for classical planning considered in the lecture that are suitable for optimal planning and two heuristics for classical planning considered in the lecture that are not suitable for optimal planning, but suitable for satisficing planning. Justify your answer (1–2 sentences).

Question 3 (6+2+2 marks)

(a) Which of the following statements about delete relaxation heuristics are generally true? For true statements, give a short (1–2 sentences) justification. For false statements, provide a counterexample.

1. $h^{\max}(s) \leq h^{\text{add}}(s)$
2. $h^{\text{FF}}(s) \leq h^{\text{add}}(s)$
3. $h^{\text{FF}}(s) \leq h^{+}(s)$

(b) Specify a family $\Pi_n = \langle V_n, I_n, O_n, \gamma_n \rangle$ ($n \in \mathbb{N}_1$) of *unit-cost, precondition-free* (i.e., *cost*$(o) = 1$ and *pre*$(o) = \top$ for all $o \in O_n$) STRIPS planning tasks such that:

$h^{\max}(I_n) = 1$ and $h^{+}(I_n) = n$ for all $n \in \mathbb{N}_1$

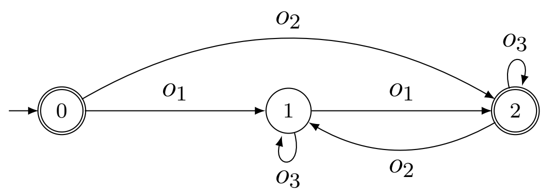
(c) Specify a family $\Pi_n = \langle V_n, I_n, O_n, \gamma_n \rangle$ ($n \in \mathbb{N}_1$) of *unit-cost, precondition-free* (i.e., *cost*$(o) = 1$ and *pre*$(o) = \top$ for all $o \in O_n$) STRIPS planning tasks such that:

$h^{+}(I_n) = 1$ and $h^{\text{add}}(I_n) = n$ for all $n \in \mathbb{N}_1$

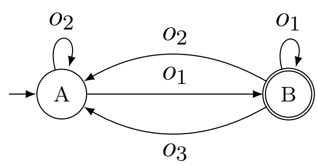
Question 4 (Abstraction heuristics, 5+3+2 marks)

(a) Specify the synchronized product $\mathcal{T}_1 \otimes \mathcal{T}_2$ of the following two transition systems. You may choose between a graphical and a formal specification.

$\mathcal{T}_1$



$\mathcal{T}_2$



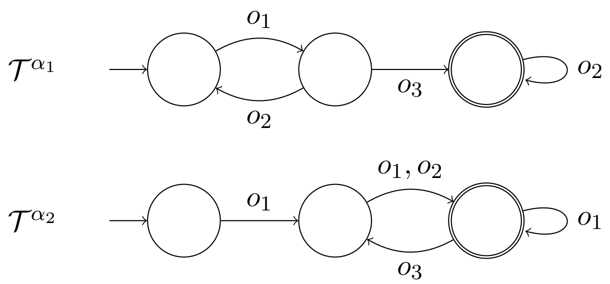
(b) Specify a *deterministic* transition system $\mathcal{T}$ with at most four states and an abstraction α of $\mathcal{T}$ such that the abstract transition system $\mathcal{T}^\alpha$ is *not deterministic*. Also draw the induced abstract system and indicate a state and action that exhibit the nondeterminism.

You may graphically specify $\mathcal{T}$ but α should be formally given.

(c) Explain how every PDB heuristic for a SAS$^+$ task can also be computed within the merge-and-shrink framework by a number of shrink steps followed by a number of merge steps.

Question 5 (Cost partitioning, 5+5 marks)

(a) Consider the following abstract transition systems $\mathcal{T}^{\alpha_1}$ and $\mathcal{T}^{\alpha_2}$ for a planning task $\Pi = \langle V, I, O, \gamma \rangle$ with $O = \{o_1, o_2, o_3\}$ and $\text{cost}(o) = 2$ for all $o \in O$.



1. (2 *marks*) Task Π is solvable. Are the abstractions α_1 and α_2 orthogonal?

   If yes, how can you best admissibly combine the estimates of h^{α_1} and h^{α_2} and what is the resulting heuristic estimate for the initial state?

   If no, explain why not.

2. (1.5 *marks*) Provide a (non-negative) cost partitioning with a heuristic value of at least 6 for the initial state. What heuristic estimate do you get for the initial state from each abstraction?

	cost o_1	cost o_2	cost o_3	heuristic estimate for I
$\mathcal{T}^{\alpha_1}$				
$\mathcal{T}^{\alpha_2}$				

3. (1.5 *marks*) Provide a general cost partitioning with a heuristic value of at least 8 for the initial state. What heuristic estimate do you get for the initial state from each abstraction?

	cost o_1	cost o_2	cost o_3	heuristic estimate for I
$\mathcal{T}^{\alpha_1}$				
$\mathcal{T}^{\alpha_2}$				

(b) Explain why the LM-cut heuristic computes a saturated cost partitioning over disjunctive action landmarks.

Address the following aspects:

- What are the landmarks?
- What costs are used for each operator when processing a specific landmark?
- Why is this cost function saturated?
- Why can we see this as a cost partitioning overall?

You may use all theorems from the lecture.