SAARLAND UNIVERSITY
Prof. Jörg Hoffmann
Dr. Peter Kissmann
Ilya Grishchenko, Evica Ilieva, Alexander Kampmann,
Vikram Mehta, Frank Nedwed, Sarah Schaeffer

First Exam
Artificial Intelligence
Summer Term 2013, July 29

Name: ....

E-mail: ....

Matr.Nr.: ....

- Write your name on every sheet of paper you use.
- Write your solutions in English.
- In the table below, mark each exercise for which you submit a solution.
- Attach all pages of your solution to these exercises.

Good luck!

Sign here:

....

Exercise	1	2	3	4	5	6	Σ
submitted							
score							
max	12	22	18	16	14	18	100

Exercise 1: Propositional Resolution (12=4+4+4 points)

For each of the following clause sets Δ, use propositional resolution to prove that Δ is unsatisfiable. That is, use propositional resolution to derive the empty clause $\Box$ from Δ. Write the resolution derivation as a graph whose nodes are clauses and whose edges lead from the parents to the resolvent of a resolution step.

(a) $\{ \{\neg P, Q\}, \{\neg Q, P\}, \{P, Q\}, \{\neg P, \neg Q\} \}$

(b) $\{ \{\neg R, Q, P\}, \{\neg P, R\}, \{\neg P, \neg Q\}, \{R, P\}, \{\neg Q\}, \{\neg P, \neg R\} \}$

(c) $\{ \{P, Q\}, \{P, R\}, \{Q, R\}, \{\neg P, \neg Q\}, \{\neg P, \neg R\}, \{\neg Q, \neg R\} \}$

Exercise 2: PL1 Normal Forms and Resolution (22=3+3+3+3+10 points)

Consider the following set of PL1 formulas:

(i) $\forall x[Dog(x) \rightarrow \exists y(Cat(y) \wedge Fast(y) \wedge Chases(x, y))]$

(ii) $\forall x, y[(Dog(x) \wedge Cat(y) \wedge Fast(y) \wedge Chases(x, y)) \rightarrow Fast(x)]$

(iii) $Dog(Lassie)$

(iv) $\neg Fast(Lassie)$

Do each of the following:

(a) Write up, in natural language (English), what these formulas mean.

(b) Bring each formula into Prenex normal form. (If the formula is already in Prenex normal form, there is nothing to do.)

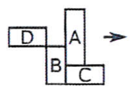
(c) Bring the formulas resulting from (b) into Skolem normal form. (If the formula is already in Skolem normal form, there is nothing to do.)

(d) Transform the formulas resulting from (c) into a set Δ of PL1 clauses. Don't forget to standardize the variables apart.

(e) Use PL1 resolution to show that Δ is unsatisfiable. Write the resolution derivation as a graph whose nodes are clauses and whose edges lead from the parents to the resolvent of a resolution step; annotate each edge with the substitution applied to unify the resolution literals (where non-empty).

Exercise 3: Heuristic Search (18 points)

Consider the Rush Hour problem given below. Each of the four rectangles represents a car parked in a parking lot denoted by the gray grid. Each car can only be moved forward or backward (i.e., cars A and B can only move up or down, car C and car D only left or right). Whenever a car is moved, it moves as far as possible, i.e., either until it touches the boundary of the grid or until it touches another car. The cost of all moves is uniformly 1, independent of the length of the move. No car may be moved outside the given grid. The goal is for car D to leave the parking lot, i.e., to reach the right-hand side boundary of the grid.



Perform A* Search as introduced in the lecture. As the heuristic h, take the number of cars blocking the path of car D to the right-hand side boundary of the grid; add 1 if car D is not yet at that boundary. In case of a tie (several nodes with identical $g + h$ value), follow this list in the given order until the ties are resolved:

1. expand those nodes with smaller h value first;
2. in case of further ties, expand states where car D is further to the right first;
3. in case of further ties, expand states with car B further down first;
4. in case of further ties, expand the states with car C further left first.

Notate the states graphically like in the figure above, and annotate each search node with its g value, its h value, its $f = g + h$ value, as well as the number n in the expansion order (in case the node is never expanded use "$n =$–"). Connect nodes by drawing edges from each node n to its child nodes (precisely, to the child nodes added to the frontier when expanding n).

Exercise 4: Adversarial Search (16=8+8 points)

Consider the Tic-Tac-Toe game after three moves by both players, as depicted in the following figure.

		X
O	O	X
X		O

X is to move and thus is the MAX-player. The goal utilities are -1 if there is a line of three Os, 0 if the board is filled without any line of three Os or Xs, and $+1$ if there is a line of three Xs.

In the following, you will have to run two search algorithms on this example. For both algorithms, always prefer marking the top-most empty cell first. If there is more than one top-most empty cell, then prefer marking the left-most of those cells first.

(a) Perform Minimax search. Draw the search tree, writing down the states in the notation as above, and connecting states by drawing edges from each state to its successor states. Annotate each state with the calculated Minimax value.

(b) Perform Alpha-Beta search. Draw the search tree as above, and mark the moves that are pruned (if any). Annotate each terminal state with its utility and each non-terminal state with "v; $[a, b]$". Here, v is the final value for that state (v in the algorithm), a is the final α value stored in that state, and b is the final β value stored in that state.

Exercise 5: STRIPS Formulations (14 points)

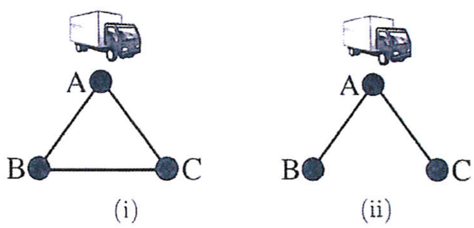
Formulate the Towers of Hanoi problem (we give a description below) with three disks and three pillars in the STRIPS formalism. Give, in STRIPS syntax, the initial state and goal state, and the set of actions. Write the actions in parameterized format, i.e., like "$move(x, y, z)$ where $x \in \{\dots\}$ and $y, z \in \{\dots\}$" instead of "$move(disc1, disc2, pillar1)$, $move(disc1, disc2, pillar2)$, $\dots$".

In the Towers of Hanoi problem, we are given a number of disks and a number of pillars. The disks are numbered from 1 to n (here, $n = 3$), and the numbers correspond to the disks' sizes. Each disk can only be placed on a larger disk, or on a pillar, if that is empty. Initially, all disks are stacked on the first pillar, and the goal is to move them to the last pillar (here we have three pillars).

Use the following predicates: *on(A,B)*, denoting that disk A is placed on top of disk or pillar B; *clear(A)*, denoting that no disk is placed on top of disk or pillar A; and *smaller(A,B)*, denoting that disk A is smaller than disk or pillar B. Also, model the following action: *move(A,B,C)*, denoting that disk A is moved from disk/pillar B to disk/pillar C.

Exercise 6: Ignoring Delete Lists (18=2+7+2+7 points)

Consider the following planning tasks, where a truck needs to visit locations B and C:



In each task, the truck has a fuel tank containing 2 fuel units, and each truck move consumes one of these fuel units. The tasks are formalized in STRIPS as follows.

For both (i) and (ii), the facts are $P = \{at(A), at(B), at(C), v(A), v(B), v(C), f(1), f(2)\}$; the initial state is $I = \{at(A), v(A), f(1), f(2)\}$; and the goal is $G = \{v(B), v(C)\}$.

Also for both (i) and (ii), the actions follow the scheme

$$drive(x, y, z) : (\{at(x), f(z)\}, \{at(y), v(y)\}, \{at(x), f(z)\})$$

The only difference between (i) and (ii) are, as depicted, the roads available, and thus the sets of concrete actions. In task (i), we have $drive(x, y, z)$ for all $(x, y) \in \{(A, B), (B, A), (A, C), (C, A), (B, C), (C, B)\}$ and $z \in \{1, 2\}$. In task (ii), we have $drive(x, y, z)$ for all $(x, y) \in \{(A, B), (B, A), (A, C), (C, A)\}$ and $z \in \{1, 2\}$.

(a) Consider task (i). Give a shortest plan for the initial state; if a plan does not exist, explain why. Give a shortest relaxed plan for the initial state: if a relaxed plan does not exist, explain why. What are the values of $h^*(I)$ and $h^+(I)$?

(b) Consider task (i). Draw the graph of all states reachable from the initial state in at most 2 steps. Writing up each state in STRIPS notation, start at the initial state, and insert successors. Indicate successor states by directed edges from the parent to the child (if two parents have the same child, draw the child only once, with two incoming edges). Annotate every state s with the values of $h^*(s)$ as well as $h^+(s)$.

(c) Consider task (ii). Give a shortest plan for the initial state; if a plan does not exist, explain why. Give a shortest relaxed plan for the initial state; if a relaxed plan does not exist, explain why. What are the values of $h^*(I)$ and $h^+(I)$?

(d) Consider task (ii). Draw the graph of all states reachable from the initial state in at most 2 steps. Writing up each state in STRIPS notation, start at the initial state, and insert successors. Indicate successor states by directed edges from the parent to the child (if two parents have the same child, draw the child only once, with two incoming edges). Annotate every state s with the values of $h^*(s)$ as well as $h^+(s)$.