POLITECNICO MILANO 1863

ARTIFICIAL INTELLIGENCE

August 26, 2021

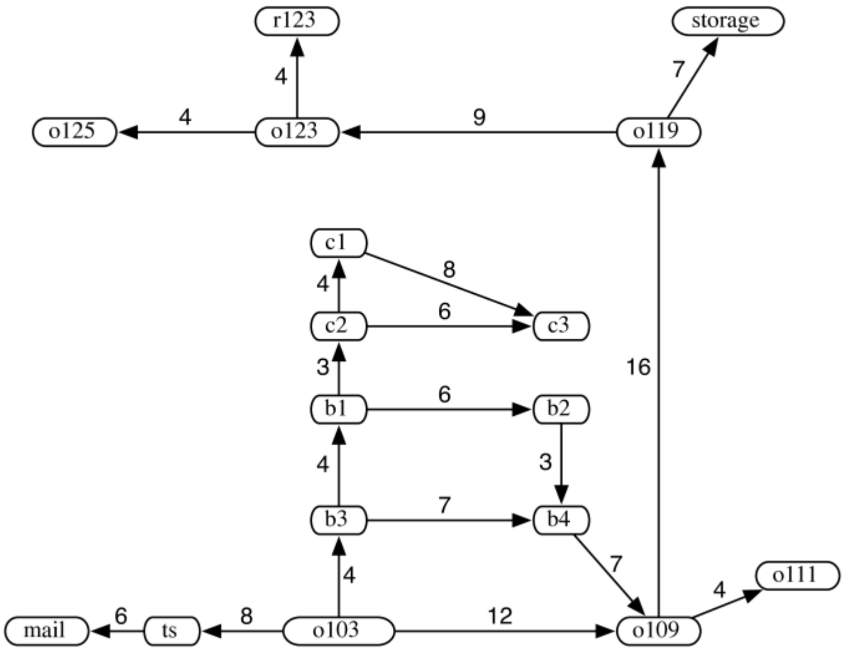
		LAST NAME AND FIRST NAME
ROW	COLUMN	ID NUMBER (CODICE PERSONA)

- The exam is composed of three stapled paper sheets printed on both sides.
- This front page must be filled with last name, first name, ID number, position (row and column communicated by the instructor), and signature.
- Exams without a completely filled front page or with missing sheets will not be considered.
- Answers can be written only on these sheets. If you need more space, please write on the last page.
- The exam is open books.
- All the answers must be justified.

SIGNATURE

Question 1 (8 points). Search.

Consider the search problem corresponding to the state space reported in the figure, in which o103 is the initial state and r123 is the (unique) goal state. Numbers on the arcs are the step costs. Assume that we want to find *all the solutions* of the search problem, so we use a modified TREE-SEARCH algorithm (without elimination of repeated states) that continues search after a solution is found.



Answer the following questions.

1.1 (2 points). What should be the termination condition for the modified TREE-SEARCH algorithm? Why?

1.2 (2 points). Can depth-first search strategy find all the solutions for the search problem above? Why?

1.3 (2 points). Can iterative-deepening search strategy find all the solutions for the search problem above? Why?

1.4 (2 points). Does A* search strategy with a consistent heuristic function $h()$ help in finding quickly all the solutions for the search problem above? Why?

1.1 The algorithm stops when the frontier is empty, which amounts to say that all the possible paths have been explored.

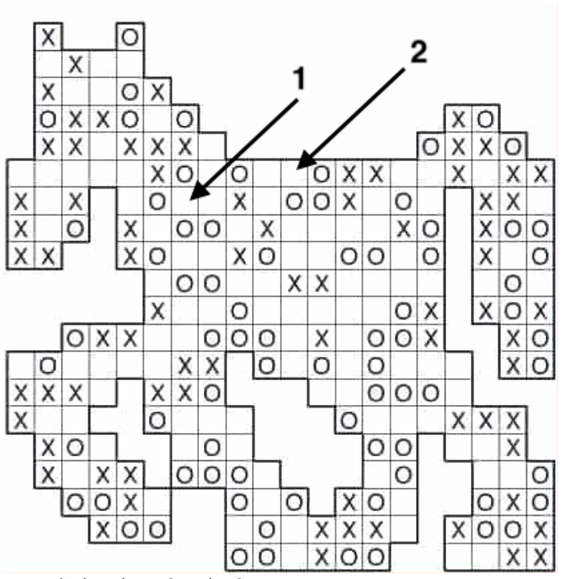
1.2 Yes, because the state space has no loops and depth-first will eventually explore all the possible paths.

1.3 Yes, see 1.2.

1.4 In general no, because the heuristic function brings the search towards the closest solution and this does not help in finding the other solutions.

Question 2 (8 points). Constraint Satisfaction Problems (CSPs).

Consider the "never-4" puzzle, which is played on a grid (like that represented in the figure) and in which the goal is to place one of two symbols (X or O) into each cell, such that there are not any horizontal, vertical, nor diagonal sequence of 4 (or more) symbols of the same type. For instance, a vertical sequence X-X-X-X and a diagonal sequence O-O-O-O-O are illegal.



2.1 (4 points). Represent a never-4 puzzle as a CSP, reporting the set of variables, the corresponding domains, and the constraints. Be as detailed as possible when describing the constraints, specifying the variables that are involved and how they are constrained.

2.2 (2 points). Assuming that forward checking is applied, what are the possible symbols that can be inserted in the cells pointed by arrows 1 and 2 in the figure? Why? If some symbols cannot be inserted in the two cells, specify when this fact has been discovered by forward checking.

2.3 (2 points). Considering the specific CSP defined in 2.1, does the application of AC-3 algorithm further reduce the domains with respect to forward checking? Why?

2.1 Variables: cells of the grid.
Domains are equal for all the variables: {X, O}.
Constraints:
(a) given any sequence of 4 horizontally-aligned cells, their values cannot be all equal,
(b) given any sequence of 4 vertically-aligned cells, their values cannot be all equal,
(c) given any sequence of 4 diagonally-aligned cells, their values cannot be all equal.

2.2 Cell 1: only O is left in its domain, since the X has been eliminated when the last of the three diagonal Xs has been inserted, due to one of the constraints (c). Cell 2: both X and O are left in its domain.

2.3 In principle, AC-3 could further reduce the domains, but the constraints in 2.1 are not binary, so AC-3 cannot be applied in this case.

Question 3 (8 points). Logic.

Consider the two following inferences, both of which are logically valid:

A:	A1.	Every student has a cellphone.
	A2.	If every student has a cellphone, then every student can be contacted.
	A3.	Therefore, every student can be contacted.
B:	B1.	Every student has a cellphone.
	B2.	Alice is a student.
	B3.	Therefore, Alice has a cellphone.

3.1 (2 points) Represent both A and B in First Order Logic (FOL).
3.2 (2 points) Now translate both A and B into Propositional Logic (PL).
3.3 (4 points) Consider the two inferences translated into PL.
- Can A3 (in PL) still be inferred from A1 and A2 (in PL)? (Justify your answer.)
- Can B3 (in PL) still be inferred from B1 and B2 (in PL)? (Justify your answer.)

FOL:	A1.	$\forall x(S(x) \to \exists y(CP(y) \land H(x,y))$	
	A2.	$\forall x(S(x) \to \exists y\big(CP(y) \land H(x,y)\big) \to \ \forall x(S(x) \to C(x))$	
	A3.	$\forall x(S(x) \to C(x))$	
	B1.	$\forall x(S(x) \to \exists y(CP(y) \land H(x,y))$	
	B2.	$S(A)$	
	B3.	$\exists y(CP(y) \land H(A,y))$	
PL:	A1.	P	where P = $\forall x(S(x) \to \exists y(CP(y) \land H(x,y))$
	A2.	$P \to Q$	where Q = $\forall x(S(x) \to C(x))$
	A3.	Q	
	B1.	P	where P = $\forall x(S(x) \to \exists y(CP(y) \land H(x,y))$
	B2.	Q	where Q = $S(A)$
	B3.	R	where R = $\exists y(CP(y) \land H(A,y))$

A3 can still be inferred from A1 and A2 in PL, e.g. via Modus Ponens

B3 cannot be inferred from B1 and B2 in PL, since the logical relationship between *S(x)* and *S(A)* is lost in PL.

Question 4 (8 points). Planning.

Consider the following PDDL action schemes, that allow a robot to pickup certain objects (O1, O2, …, On) from a table and put them into a number of boxes (B1, B2, …, Bm):

```
action pickup(x)  // pick up object x from the table
  prec   …
  eff    …
action put(x,y)  // put object x into box y
  prec   …
  eff    …
```

A typical initial state, where all boxes are empty, will have the form:

Object(O1), …, Object(On), OnTable(O1), …, OnTable(On), Box(B1), …, Box(Bm), Free

4.1 (3 points) Complete the action schemes above specifying preconditions and effects (you can define new predicates, if needed).

4.2 (2 points) Then represent the following possible goals, assuming that your version of PDDL allows for the use of equality (=) and inequality (≠) between variables and/or constants:

G1. At least one object is in a box (note: which particular object and which particular box is NOT specified)

G2. At least two different boxes have at least one object each (note: which particular boxes have an object is NOT specified)

4.3 (3 points) Finally suppose that we want to represent the further goal:

G3. There are no objects left on the table

in such a way that G3 is independent of the particular set of objects that are specified in the initial state. Explain why the goal:

Object(x) ∧ ¬OnTable(x)

does not serve the purpose.

```
action pickup(x)  // pick up object x from the table
  prec   Object(x) ∧ OnTable(x) ∧ Free
  eff    ¬OnTable(x) ∧ ¬Free ∧ Has(x)

action put(x,y)  // put object x into box y
  prec   Has(x) ∧ Box(y)
  eff    ¬Has(x) ∧ InBox(x,y) ∧ Free
```

G1. InBox(x,y)

G2. InBox(x1,y1) ∧ InBox(x2,y2) ∧ (y1 ≠ y2)

G3. Object(x) ∧ ¬OnTable(x) does not serve the purpose because it is satisfied as soon as an object is picked up by the robot.