WeST People and Knowledge Networks

UNIVERSITÄT KOBLENZ · LANDAU

Artificial Intelligence 1 SS 2022

Demo Exam

- You do not have to submit the demo exam.
- You will find the sample solution in OLAT.
- You can ask questions about the sample solution in the tutorial on July 28 and 29, 2022.

1 Classical logics (20 Points)

Formulate the following sentences in first-order logic. Use the following predicates:

- *offers(P,O)* to indicate that person P offers the food option O.
- *likes(P,O)* to indicate that person P like food option O.
- *isVegan(O)* to indicate that O is vegan.

Furthermore, the constants *pizza*, *curry*, *sandwich* and *pasta* represent food options whereas *alice* and *bob* denote individual persons.

1. Alice likes either Pizza or Sandwiches.

   Formula: ____

2. Bob likes only vegan food.

   Formula: ____

3. Everyone who offers Pizza also offers Pasta.

   Formula: ____

4. There is a person who offers Pasta but does not like Curry.

   Formula: ____

5. If a person only likes vegan food, then he offers Curry or Sandwiches but not both.

   Formula: ____

Solution

1. $(likes(alice, pizza) \vee likes(alice, sandwich)) \wedge \neg(likes(alice, pizza) \wedge likes(alice, sandwich))$
2. $\forall X : (likes(bob, X) \Rightarrow isVegan(X))$
3. $\forall X : (offers(X, pizza) \Rightarrow offers(X, pasta))$
4. $\exists X : (offers(X, pizza) \wedge \neg likes(X, curry))$
5. $\forall X : \forall Y((likes(X, Y) \Rightarrow isVegan(Y)) \Rightarrow ((offers(X, curry) \wedge \neg offer(X, sandwich)) \vee (\neg offers(X, curry) \wedge offers(X, sandwich)))$

Let Γ, Δ, Θ and Λ be sets of propositional formulae. For each of the following statements, say whether it is true or false. Please note: For each correct cross you get 1 points. For each incorrect cross 1 points are subtracted. In total, there is never less than 0 points for this subtask.

true	false	
☐	☐	$\{a \wedge b, c \vee \neg b\} \vdash a \vee c$.
☐	☐	$\{a, a \vee b, c \wedge \neg d\} \vdash a \wedge d$.
☐	☐	$\{c, a \vee \neg b, a \wedge b, \neg a \wedge c\} \vdash a \vee c$.
☐	☐	$\{a \wedge b, \neg b \wedge c, c \wedge \neg a\}$ is satisfiable.
☐	☐	If $\Gamma \cap \Delta \vdash \psi$ and $\Theta \cup \Lambda \vdash \psi$ than $\Delta \cup \Lambda \vdash \psi$.

Solution

true - false - true - false - true

Explain why

$$\forall X : (likes(alice, X) \Rightarrow offers(alice, X)) \nvdash \exists X : (likes(alice, X) \wedge offers(alice, X)).$$

Do this by finding an interpretation I that is a model of $\forall X : (likes(alice, X) \Rightarrow offers(alice, X))$ but not of $\exists X : (likes(alice, X) \wedge offers(alice, X))$. Remember to state all necessary parts of the tuple $I = (U_I, f_I^U, P_I, F_I)$.

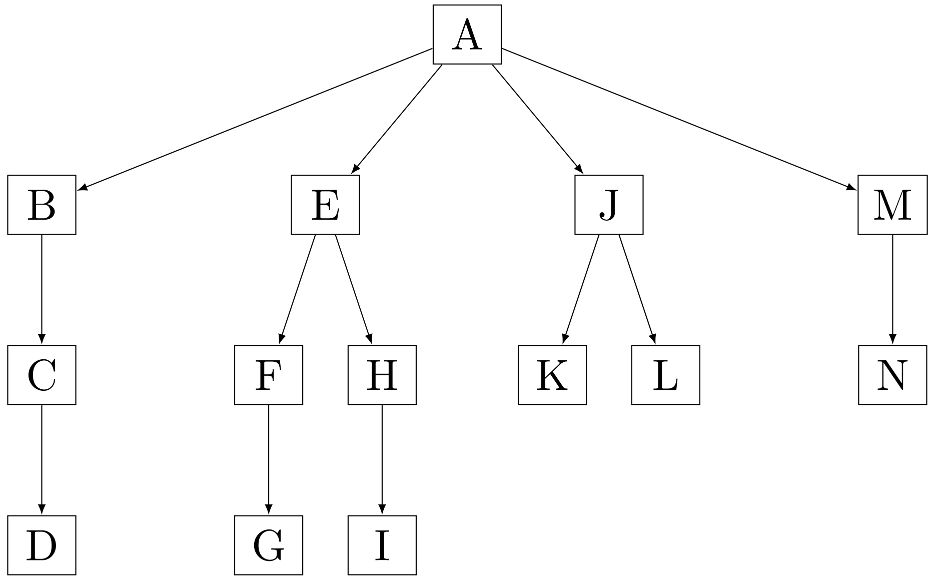
Solution

E.g. the model $I = (U_I, f_I^U, P_I, F_I)$ defined by:

- $U_I = \{a\}$
- $f_I^U(alice) = a$
- $P_I = \{likes^I, offers^I\}$ with $likes^I = \emptyset$, $offers^I = \emptyset$
- $F_I = \emptyset$

2 Uninformed Search (15 Points)

Consider the following search tree of an uninformed search problem:



In which order do the following algorithms explore the nodes of the search tree, if the start node is A and the target node is K? In case of nodes being added to the `Frontier` at the same time, you should process them in alphabetical order.

1. Breadth-First-Search:

   Exploration order:

2. Depth-First-Search:

   Exploration Order:

3. Name two advantages of Breath-First-Search over Depth-First-Search.

Solution

1. Breadth First Search:

   Exploration order: A - B - E - J - M - C - F - H - K

2. Depth First Search:

   Exploration order: A - B - C - D - E - F - G - H - I - J - K

3. Breath: complete and optimal

3 Search Problems (20 Points)

The *tower of hanoi* is a puzzle consisting of three rods and a number of disks (in our case three) of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on the left rod, the smallest disk at the top, thus making the shape of a cone (depicted in Figure 1).

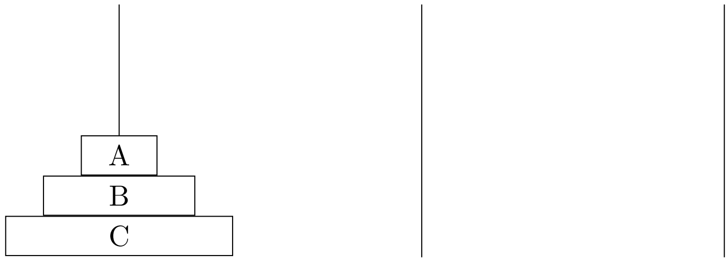


Figure 1: Starting configuration of tower of hanoi

The objective of the puzzle is to move the entire stack to the rightmost rod, obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or an empty rod.
- No larger disk may be placed on top of a smaller disk.

a) Define the search problem $P = (S, O, I, G)$ where S is the set of states, O the transition relation, I the initial state and G the set of goal nodes. For the transition relation, it is enough to give three examples of members of said relation.

b) Define an admissible heuristic h_1 for the problem. Say whether your heuristic is monotonous.

c) Assume that your heuristic h_1 is admissible. Furthermore, assume a second admissible heuristic h_2 is given and that h_2 is different from h_1 but just as easy to compute. Say whether $h_3 = max(h_1, h_2)$ is:

	true	false	
Please note:	☐	☐	h_3 is admissible.
	☐	☐	h_3 dominates h_2.
	☐	☐	Using A* with h_3 is as least as fast as using A* with h_1.

Solution

Search problem P is $P = (S, O, I, G)$. S list of list $S = [[L], [M], [R]]$ with $L \cup M \cup R = \{A, B, C\}$ and $L \cap M \cap R = \emptyset$.

- $[[A, B, C], [], []] \rightarrow [[B, C], [A], []] \in O$
- $[[A, B, C], [], []] \rightarrow [[B, C], [], [A]] \in O$
- $[[B, C], [A], []] \rightarrow [[C], [A], [B]] \in O$

$I = [[A, B, C], [], []]$

$G = [[], [], [A, B, C]]$

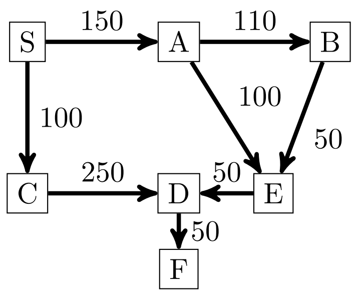
b) One heuristic is the number of misplaced disks. This heuristic is monotonic because every misplaced disk has to be moved atleast once.

c) $h3$ is admissible - $h3$ dominates h_2 - h_3 is preferable (so A* with h_3 is as least as fast as with h_1).

4 Informed Search (15 Punkte)

Assume the following graph. Your starting node is S. The target node is F. The table contains an estimation of the cost from each node to the goal node.

Knoten	Estimated Distance to F
S	300
A	200
B	100
C	100
D	25
E	25
F	0



Use the A*-Algorithm to find the shortest path between S and F. Use one row of the following table for each step, starting with step 2. Always write down the node (together with the path to the node) and its value according to the state evaluation function f. If there are two nodes with the same value for f in the frontier, select the one which was added last to the frontier. Circle the node that is being taken out of the Frontier in each step. Write down the path that the algorithm found. Also state whether the path is optimal.

Step	Frontier			
1	S = 300			
2	SA=	SC =		
3				
4				
5				
6				
7				
8				

Order in which the nodes are taken from the frontier: ____________________

Found path: ____________________

Costs of the path: ____________________

Is the found path optimal? Yes ☐ No ☐

Solution

Schritt	Frontier		
1	**S = 300**		
2	SA = 150 + 200 = 350	**SC = 100 + 100 = 200**	
3	**SA = 350**	SCD = 350 + 25 = 375	
4	SAB = 250 + 110 = 360	SCD = 375	**SAE = 250 + 25=275**
5	SAB = 360	SCD = 375	**SAED = 300 + 25=325**
6	SAB = 360	SCD = 375	**SAEDF = 350 + 0=350**

Order of nodes taken from the frontier: $SCAEDF$
The found path is $SAEDF$. This path with cost 350 is optimal.

5 STRIPS (10 Points)

The purpose of the *mail collector robot* is to retrieve the mail from the mailbox in the frontyard. The robot is located in the hallway. There is also a cupboard in the hallway. The aim is to put the mail in this cupboard. The cupboard furthermore contains a key that opens the front door. This key should be in the cupboard after picking up the mail. To retrieve the mail, the robot needs to perform the following actions:

- pick up the key from the cupboard
- open the front door
- put the key back in the cupboard
- go out into the front yard where the mailbox is
- grab the mail from the mailbox
- go back inside
- put the mail in the cupboard

To model this we use the following constants to denote the objects in our world: *hallway* and *frontyard* for the two locations, *door* for the front door, *key* for the key, *robot* for the robot, *cupboard* for the cupboard, *mailbox* for the mailbox, and *mail* for the mail.

We use the following predicates to represent the state of our world:

- $\mathit{location}(L)$: L is a location.
- $\mathit{door}(D)$, $\mathit{key}(K)$, $\mathit{mail}(M)$: D is a door, K is a key, M is mail.
- $\mathit{container}(C)$: C is a container (e.g. *mailbox* or *cupboard*).
- $\mathit{connects}(D, L_1, L_2)$: Door D connects the two locations L_1 and L_2.
- $\mathit{open}(D)$: D is open.
- $\mathit{at}(T, L)$: The item T (box or robot) is at location L.
- $\mathit{contains}(C, T)$: C (e.g. *mailbox* or *cupboard*) contains item T (e.g. *key* or *mail*).
- $\mathit{holds}(T)$: The robot carries item T.

There are two possible actions: $\mathit{move}(X, Y)$ and $\mathit{pickup}(T)$. The $\mathit{move}(X, Y)$ action moves the robot from location X to Y. This is only possible if the robot is at location X, if Y is also a location, and if X and Y are connected through a door and this door is open.

The action $\mathit{pickup}(T)$ makes the robot pick something up from a container. This is only possible if T is contained in a container that is at same location as the robot.

Some further actions could be modelled, such as *putdown*, *unlock* or *open*, but we will ignore these.

a) Describe the initial state and the goal using the predicates and constants above.

Database: ____

____

Goal: ____

b) Describe the actions $move(X, Y)$ and $pickup(T)$ using STRIPS notation by specifying the action's (pre-)conditions C, delete-list D, and add list A:

move(X,Y): C: ________________

D: ________________

A: ________________

pickup(T): C: ________________

D: ________________

A: ________________

Solution

a) Database: $\{at(robot, hallway), at(cupboard, hallway), at(mailbox, frontyard),$
$connects(door, hallway, frontyard), connects(door, frontyard, hallway), contains(mailbox, mail),$
$contains(cupboard, key), container(cupboard), container(mailbox), door(door), key(key), mail(mail)\}$.
Goal: $\{contains(cupboard, mail), contains(cupboard, key)\}$.

b) move(X,Y):
C: at(robot,X), connects(D,X,Y), door(D), location(X), location(Y), open(D)
D: at(robot, X)
A: at(robot, Y)

pickup(T):
C: at(robot, X), container(C), at(C,X), contains(C,T)
D: contains(C,T)
A: holds(T)

6 Default theory (20 Punkte)

Let $T = (W, \Delta)$ be a default theory with $W = \emptyset$ and $\Delta = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5\}$ with

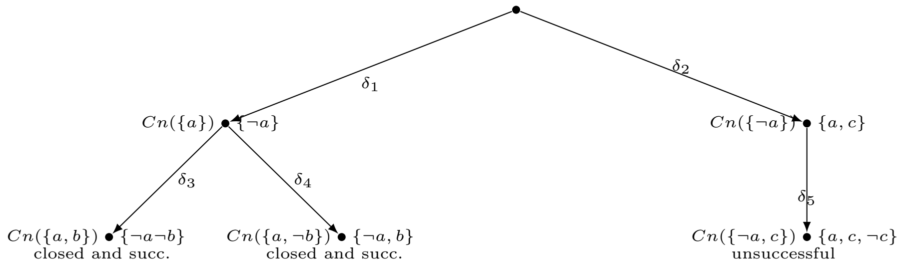
$$\delta_1 = \frac{\top \ : \ a}{a}, \quad \delta_2 = \frac{\top \ : \ \neg a, \ \neg c}{\neg a}, \quad \delta_3 = \frac{a \ : b}{b}, \quad \delta_4 = \frac{a \ : \ \neg b}{\neg b}, \quad \delta_5 = \frac{\neg a \ : \ c}{c}$$

Calculate all extensions of T using a process tree.

$\mathsf{Cn}(W) \quad \bullet \quad \emptyset$

Extensions: ____________________

Solution



Extensions are: $Cn(\{a, b\}), Cn(\{a, \neg b\})$

7 Answer set programming (20 Points)

Consider the following extended logic program P:

$$
\begin{aligned}
P: \quad & a. \\
& b \leftarrow \neg q, not\ e. \\
& d \leftarrow not\ p. \\
& p \leftarrow not\ d. \\
& e \leftarrow not\ c. \\
& c \leftarrow a, b.
\end{aligned}
$$

a) Find all answer sets of P. For each answer set, provide the reduct and justify why it is an answer set. *Hint*: There are exactly two answer sets.

b) Transform the extended logic program into a default theory such that there is a one to one correspondence of answer sets and extensions.

Solution

There are two models:

- $S_1 = \{a, p, e\}$, the reduct

$$
\begin{aligned}
P^{S_1} :& a. \\
& p. \\
& e. \\
& c \leftarrow a, b.\}
\end{aligned}
$$

  and the minimal model $minModel(P^{S_1}) = S_1 = \{a, p, e\}$.

- $S_2 = \{a, d, e\}$, the reduct

$$
\begin{aligned}
P^{S_1} :& a. \\
& d. \\
& e. \\
& c \leftarrow a, b.\}
\end{aligned}
$$

  and the minimal model $minModel(P^{S_2}) = S_2 = \{a, d, e\}$.

An equivalent default theory is $T = (W, \Delta)$ with $W = \emptyset$ and $\Delta = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7\}$ with

$$
\delta_1 = \frac{\top\ :}{a}, \quad \delta_2 = \frac{\neg q\ :\ \neg e}{b}, \quad \delta_3 = \frac{\top\ :\ \neg p}{d}, \quad \delta_4 = \frac{\top\ :\neg d}{p} \quad \delta_5 = \frac{\top\ :\neg c}{e} \quad \delta_6 = \frac{a \wedge b\ :}{c}
$$

8 Prolog (10 Points)

a) Consider the following Prolog database:

```
p(a,b).
p(b,d).
p(e,b).
p(b,f).
p(a,c).
p(c,f).
p(h,g).
p(h,c).
q(X,Y) :- p(X,Y).
q(X,Z) :- p(X,Y), q(Y,Z).
```

Given this database, what does the following query return? (If there are multiple answers then list them all)

```
?- q(a,X).
```

b) Now consider the following Prolog database:

```
a(1).
a(2).
b(1).
b(2).
c(2).
d(X) :- a(X), b(X), !, c(X).
```

Given this database, what does the following query return? (If there are multiple answers then list them all)

```
?- d(X).
```

Solution

a)
```
X = b
X = c
X = d
X = f
X = f
```

b) The query returns `false`.