Homework 5 – Artificial Intelligence

Teacher: Stefán Ólafsson

March 10, 2023

Time Estimate

4 hours including reading the relevant parts of chapters 7 (7.1-7.5) and 11 (11.1-11.2), assuming you have been to the lectures on logic and planning or watched the recordings.

Questions

1. Logic (30 points)

(Based on "The Adventure of Silver Blaze", an original Sherlock Holmes mystery by Arthur Conan Doyle)

A prize-winning racehorse named Silver Blaze has been stolen from a stable, and a bookmaker named Fitzroy Simpson has been arrested as the prime suspect by good old Inspector Gregory. Sherlock Holmes, however, after ample use of his magnifying glass and some of the strongest black tobacco this side of the Atlantic, finds the true thief by reasoning from the following premises:

- The horse was stolen either by Fitzroy or by its trainer John Straker.
- The thief had to have entered the stable the night of the theft.
- If a stranger enters the stable, the dog barks.
- Fitzroy was a stranger.
- The dog did not bark.

Who stole Silver Blaze?

We can model the problem using propositional logic by introducing the following propositional symbols:

symbol	meaning
TF	Fitzroy is the thief
TJ	John is the thief
EF	Fitzroy entered the stable
EJ	John entered the stable
SF	Fitzroy is a stranger
SJ	John is a stranger
DB	the dog barked

Based on these symbols, we can encode the sentences above in propositional logic, resulting in the following knowledge base:

#	sentence	explanation
1	$TF \vee TJ$	The horse was stolen by Fitzroy or John,
2	$\neg(TF \wedge TJ)$	but not by both (either ... or).
3	$\neg SJ$	Implicit in the text: The trainer is not a stranger.
4	$TF \Rightarrow EF$	If someone is a thief, they must have entered
5	$TJ \Rightarrow EJ$	the stable the night of the theft.
6	$EF \wedge SF \Rightarrow DB$	If a stranger enters the stable,
7	$EJ \wedge SJ \Rightarrow DB$	the dog barks.
8	SF	Fitzroy was a stranger.
9	$\neg DB$	The dog did not bark.

Your task is to use the inference rules and equivalences below to infer who is the thief. That is, starting with the given knowledge base, apply inference rules to infer additional sentences until you infer either TF or TJ. For each inference step note which sentences and which inference rule / equivalence you used, for example, in this form:

#	inference rule	inferred sentence
1-9	from knowledge base	
10	rule 4 with sentences 3 and 8	$\neg SJ \wedge SF$
11	...	...

Inference rules

$$\{\alpha \Rightarrow \beta, \alpha\} \vdash \beta \quad (1)$$
$$\{\alpha \Rightarrow \beta, \neg\beta\} \vdash \neg\alpha \quad (2)$$
$$\{\alpha \wedge \beta, .\} \vdash \alpha \quad (3)$$
$$\{\alpha, \beta\} \vdash \alpha \wedge \beta \quad (4)$$
$$\{\alpha, .\} \vdash \alpha \vee \beta \quad (5)$$
$$\{\alpha \vee \beta, \neg\alpha\} \vdash \beta \quad (6)$$
$$\alpha \Leftrightarrow \beta \equiv \beta \Leftrightarrow \alpha \quad (7)$$
$$\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge \beta \Rightarrow \alpha \quad (8)$$
$$\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta \quad (9)$$
$$\alpha \wedge \beta \equiv \beta \wedge \alpha \quad (10)$$
$$\alpha \vee \beta \equiv \beta \vee \alpha \quad (11)$$
$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta \quad (12)$$
$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta \quad (13)$$
$$\neg\neg\alpha \equiv \alpha \quad (14)$$

Equivalences ($\equiv$) can be used as inference rules in both directions. E.g., $\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$ means $\{\alpha \Rightarrow \beta, .\} \vdash \neg\alpha \vee \beta$ and $\{\neg\alpha \vee \beta, .\} \vdash \alpha \Rightarrow \beta$.

2. STRIPS Planning (70 points)

You are a robot that gets the task of locking all the rooms of a building. The building has 2 rooms numbered 1 and 2. In order to lock a room you need to have the key for the room. Both keys are in room 1. You only have one hand, which can only hold one item. You have the STRIPS operators below available to you and the closed world assumption holds.

2.1 (15 points) Write down the following state constraints as first-order logic sentences using the vocabulary (i.e., the relations used to describe the state, preconditions, etc) from the STRIPS description below.

- The robot can only be at one place.
- If the robot holds something, its hand is not empty and vice versa.
- An object that is held by the robot is nowhere else.

2.2 (10 points) Estimate the branching factors for forward search and for backward search and explain your estimates. Compare the two branching factors.

2.3 (5 points) Suppose the robot could also ring alarm bells, greet anyone who enters the building by name, and make coffee. How would that change your estimates for the branching factors?

2.4 (40 points) Use **backward search** to plan your course of action. You can use the state constraints above to cut-off nodes in the search tree. Write down the solution path (e.g., goal condition $\xrightarrow{action_1}$ regressed goal_1 $\xrightarrow{action_2}$ ... $\xrightarrow{action_n}$ condition which is fulfilled in initial state).

2.5 (10 bonus points) Assume all actions have a cost of 1 and we would use best-first search with the planning graph heuristic as a forward search to find a solution to the problem. What is the cost of the solution that this algorithm would find? Explain your answer.

Action($GotoRoom(x, y)$,
Precond: $At(robot, x)$,
Add: $At(robot, y)$,
Delete: $At(robot, x)$ **).**

Action($PickUp(o, x)$,
Precond: $At(robot, x) \wedge At(o, x) \wedge Handempty$,
Add: $Holding(robot, o)$,
Delete: $At(o, x) \wedge Handempty$ **).**

Action($Drop(o, x)$,
Precond: $At(robot, x) \wedge Holding(robot, o)$,
Add: $At(o, x) \wedge Handempty$,
Delete: $Holding(robot, o)$ **).**

Action($Lock(x)$,
Precond: $At(robot, x) \wedge Holding(robot, key(x))$,
Add: $Locked(x)$,
Delete: $-$ **).**

Action($Unlock(x)$,
Precond: $At(robot, x) \wedge Holding(robot, key(x))$,
Add: $-$,
Delete: $Locked(x)$ **).**

Initial State:
$At(robot, 1) \wedge Handempty \wedge At(key(1), 1) \wedge At(key(2), 1)$

Goal Condition:
$Locked(1) \wedge Locked(2) \wedge At(key(1), 1) \wedge At(key(2), 1)$

Figure 1: STRIPS description of the security robot domain.