Einführung in Künstliche Intelligenz SS 2019, 2.0 VU, 184.735

Exercise Sheet 3 – CSPs, Planning, and Making Simple Decisions

For the discussion part of this exercise, mark your solved exercises in **TUWEL** until Sunday, June 09, 23:55 CEST. The registration for a solution discussion ends on Friday, June 14, 23:55 CEST. Be sure that you tick only those exercises that you can solve and explain!

In the discussion, students will be asked questions about their solutions of examples they checked. The discussion will be evaluated with 0-15 points, which are weighted with the fraction of checked examples and rounded to the next integer. There is *no minimum number of points* needed for a positive grade (i.e., you do not need to participate for a positive grade, but you can get at most 85% without doing exercises).
Note, however, that *your registration is binding*. Thus, *if* you register for a solution discussion, then it is *mandatory* to show up. Not coming to the discussion after registration will lead to a reduction of examination attempts from 4 to 2.
Please ask questions in the **TISS** Forum or visit our tutors during the tutor hours (see **TUWEL**).

Exercise 3.1: Consider the following cryptarithmetic puzzle. Every letter corresponds to exactly one digit. In particular, the digits corresponding to different letters are different. Furthermore, A and G should not be 0, and W must be greater than I.

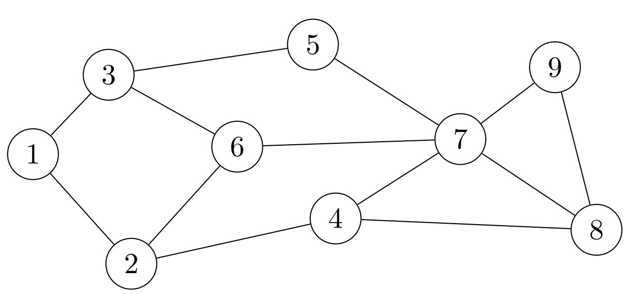
```
  W I T C H
+   W A N D
-----------
  M A G I C
```

(a) Describe the corresponding CSP with its variables and constraints and specify the initial domain of each variable.

(b) Draw the constraint graph.

(c) Find a solution of the puzzle. How many unique solutions exist? You may use automated solvers or program your own, but be sure that you can explain how a solution could be found by hand.

Exercise 3.2: Consider the three-colorability problem for the following graph:

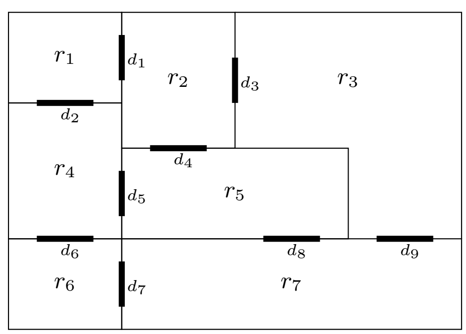


Find a valid 3-coloring for the graph using the colors red, blue, and green. Select the first node by using the *degree heuristic*. After that, select the nodes according to the *minimum remaining values heuristic* and the *degree heuristic* if the remaining values of two or more nodes are equal. If you still have multiple options after applying both heuristics, select the node with the smallest number. Furthermore, select the colors according to the *least constraining value heuristic*.

Exercise 3.3: A time-travelling secret agent arrives at Kurt Gödel's apartment in the middle of a night in 1929 to retrieve a mobile phone that her colleague had left behind while taking a selfie. The mission is of vital importance, as if she fails, the fabric of space-time will be profoundly damaged. The colleague left her a detailed plan of the house, specifying which doors he has left unlocked, and told her where the phone is located. Our agent must navigate through the house, possibly unlocking any locked doors and retrieve the lost mobile phone. If there is already an unlocked door between two rooms, she may simply pass. She may unlock any locked door if she is in one of the two rooms connected by it. As she is interested in doing this before the brilliant mind awakes, she needs to make as few moves as possible.

(a) Design two STRIPS actions, one for crossing from one room to another and one for unlocking a locked door between two rooms. Define the variables to model different aspects of this exercise on your own and describe them in detail.

(b) You are given the blueprint of the house that the agent received. Rooms are labeled with $r_1, \ldots, r_7$. Thick lines on the walls indicate a door between two rooms and are labeled with $d_1, \ldots, d_9$. The agent was told that the doors d_2, d_5, and d_9 have been left unlocked. Upon arriving in the alternate timeline, she finds herself in the room r_1, while the phone is in the room r_7. Formulate the initial state of the given planning setting and use *progression planning* to find a plan to retrieve the phone. What do the goal states look like?



Exercise 3.4: Squirrels have always been your favorite animals. Aiming to increase the squirrel population in Vienna, you have built a prototype of a robot squirrel. To make it look more realistic, the squirrel stores nuts during the summer and eats them during the winter. At the moment, the squirrel has three hiding places, H_1, H_2, and H_3. H_1 contains a small heap of acorns, H_2 contains some walnuts, while H_3 is empty. The squirrel can take all contents out of a hiding place or put them back in. You have specified the following actions:

$$\begin{aligned}
&\textbf{Action}(\textit{Take}(h, n)),\\
&\quad \text{Precond}: \textit{free} \wedge \textit{contains}(h, n) \wedge \textit{at}(h),\\
&\quad \text{Effect}: \neg\textit{free} \wedge \textit{holds}(n) \wedge \neg\textit{contains}(h, n) \wedge \textit{empty}(h))
\end{aligned}$$

$\textbf{Action}(Put(h, n)),$

Precond : $holds(n) \wedge empty(h) \wedge at(h),$

Effect : $free \wedge \neg holds(n) \wedge contains(h, n) \wedge \neg empty(h))$

$\textbf{Action}(Move(h_1, h_2)),$

Precond : $at(h_1),$

Effect : $\neg at(h_1) \wedge at(h_2))$

The meaning of the predicates is as follows:

$at(h)$: the squirrel is at hiding place h,
$contains(h, n)$: hiding place h contains nuts n,
$empty(h)$: hiding place h is empty,
free: the squirrel's arms are free,
$holds(n)$: the squirrel is holding n.

The squirrel finds itself in the initial state

$$S := \{free, at(h_3), contains(h_1, acorns), contains(h_2, walnuts), empty(h_3)\}.$$

Deciding to reshuffle the contents of the hiding places, so that the acorns are closer to its tree, the squirrel needs to reach the goal state

$$G := \{free, at(h_2), empty(h_1), contains(h_2, acorns), contains(h_3, walnuts)\}.$$

Use the STRIPS state-space search algorithm starting in S (i.e., use *progression planning*) to determine the shortest possible plan that achieves the desired goal state G.

Exercise 3.5: Assume there is a lottery with tickets for €5 and there are three possible prizes: €1000 with a probability of 0.05%, €100 with probability 0.1%, and €1 otherwise.

(a) What is the expected monetary value of a lottery ticket?

(b) When is it rational to buy a ticket? Give an inequality involving utilities with the following utilities: $U(S_k) = 0$, $U(S_{k+5}) = 5 \cdot U(S_{k+1})$, $U(S_{k+100}) = 50 \cdot U(S_{k+5})$, but there is no information about $U(S_{k+1000})$.

(c) Define $U(S_{k+5})$ and $U(S_{k+1000})$ such that a rational agent whose utility function satisfies the equations in Subtask (b) chooses to buy a lottery ticket.

Exercise 3.6: In 1713, Nicolas Bernoulli investigated a problem, nowadays referred to as the *St. Petersburg paradox*, which works as follows. You have the opportunity to play a game in which a fair coin is tossed repeatedly until it comes up heads. If the first head appears on the n-th toss, you win 2^n Euros.

(a) Show that the expected monetary value of this game is not finite.

(b) Daniel Bernoulli, the cousin of Nicolas, resolved the apparent paradox in 1738 by suggesting that the utility of money is measured on a logarithmic scale, i.e., $U(S_n) = a \log_2 n + b$, where S_n $(n > 0)$ is the state of having n Euros and a, b are constants. What is the expected utility of the game under this assumption? Assume, for simplicity, an initial wealth of 0 Euros and that no stake has to be paid in order to play the game.