

Foundations of Artificial Intelligence

M. Helmert
S. Eriksson
Spring Term 2022

University of Basel
Computer Science

Exercise Sheet 4

Due: March 27, 2022

Important: for submission, consult the rules at the end of the exercise. Non-adherence to the rules will lead to your submission not being corrected.

Exercise 4.1 (1+1 marks)

Are the following statements about depth-first search correct? Justify your answer.

- (a) For a solvable state space $\mathcal{S}_s$, there always exists an expansion order such that depth-first search terminates.
- (b) For an unsolvable state space $\mathcal{S}_u$, there always exists an expansion order such that depth-first search terminates.

Exercise 4.2 (1+1 marks)

Which of the search algorithms shown in slide 35 of the printout version of chapter 12 would you use to find optimal solutions for the following problems? For breadth-first and uniform cost also consider whether to use the tree or graph variant. Justify your answer in one to two sentences.

- (a) Brute-forcing a password of unknown length, where each action (with cost 1) consists of adding a character to the string and the goal is reached as soon as we have the correct string (no enter key needed). We assume a maximal password length of 100.
- (b) The lights-out problem as defined on exercise sheet 2.

Exercise 4.3 (2 marks)

We consider a $n \times m$ lights-out problem where states are defined as tuples $\langle c_{1,1}, \dots, c_{i,j}, \dots, c_{n,m} \rangle$ with $c_{i,j} \in \{0,1\}$, and where $c_{i,j} = 1$ denotes that a cell is on while $c_{i,j} = 0$ denotes that a cell is off. We define the heuristic

$$h(s) = \sum_{i=1}^n \sum_{j=1}^m c_{i,j}.$$

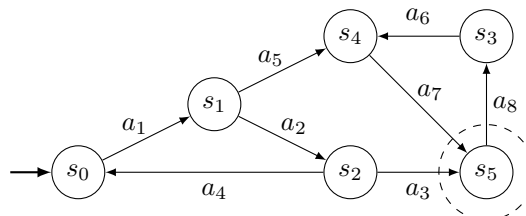
Formally prove that h is safe, goal-aware, admissible and consistent.

Hint: You may use results from the lecture slides.

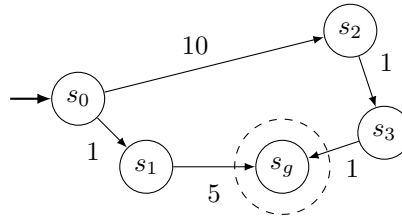
Exercise 4.4 (2 marks)

For the state space depicted below with uniform action costs of 1, define a heuristic that is consistent, not safe and has a perfect estimate for s_0 (i.e., $h(s_0) = h^*(s_0)$). Justify your answer by showing why the heuristic is consistent and not safe.

Hint: If a consistent heuristic assigns ∞ to a state, all its successors must be assigned ∞ as well.



Exercise 4.5 (1+1 marks)



Execute

- (a) A^* ($f(n) = g(n) + h(n.state)$) and
- (b) greedy best-first search ($f(n) = h(n.state)$)

in the state space depicted above. As heuristic, use the perfect heuristic h^* . Describe the execution of both search algorithms with the following schema:

1. expanding s_{10} : $open = \langle s_{11}(f = 5), s_{12}(f = 6) \rangle$, $closed = \{s_{10}(g = 0)\}$
2. expanding s_{11} : $open = \langle s_{13}(f = 4), s_{12}(f = 6), s_{16}(f = 10) \rangle$, $closed = \{s_{10}(g = 0), s_{11}(g = 3)\}$
3. ...
4. expanding s_i : found goal with cost x

Note: The schema above uses made up states and numbers that don't correspond to the given state space.

Submission rules:

- Exercise sheets must be submitted in groups of two students. Please submit a single copy of the exercises per group (only one member of the group does the submission).
- Create a single PDF file (ending .pdf) for all non-programming exercises. Use a file name that does not contain any spaces or special characters other than the underscore “_”. If you want to submit handwritten solutions, include their scans in the single PDF. Make sure it is in a reasonable resolution so that it is readable, but ensure at the same time that the PDF size is not astronomically large. Put the names of all group members on top of the first page. Either use page numbers on all pages or put your names on each page. Make sure your PDF has size A4 (fits the page size if printed on A4).
- For programming exercises, only create those code textfiles required by the exercise. Put your names in a comment on top of each file. Make sure your code compiles and test it. Code that does not compile or which we cannot successfully execute will not be graded.
- For the submission: if the exercise sheet does not include programming exercises, simply upload the single PDF. If the exercise sheet includes programming exercises, upload a ZIP file (ending .zip, .tar.gz or .tgz; *not* .rar or anything else) containing the single PDF and the code textfile(s) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly.
- Do not upload several versions to ADAM, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.