Planning and Optimization

M. Helmert, G. Röger
R. Christen, P. Ferber, T. Keller

University of Basel
Fall Semester 2022

Exercise Sheet 3

Due: October 17, 2022

Important: for submission, consult the rules at the end of the exercise. Non-adherence to these rules might lead to a penalty in the form of a deduction of marks or, in the worst case, that your submission will not be corrected at all.

Exercise 3.1 (0.5+1+1.5 marks)

This exercise is a literature research question. The goal of such exercises is to find information in research papers. We don't expect you to fully read the paper. Instead, try to extract the relevant information to answer the question. Use your own words in your answers to avoid plagiarism.

Consider the following paper to answer the questions below:

> Slaney, J. and Thiébaux, S. (2001). Blocks World revisited.
> *Artificial Intelligence*, 125(1–2), 119–153.

(a) Explain in your own words what is necessary that a block is *in position*.

(b) Provide a Blocksworld instance where the GN1 algorithm computes an optimal solution and the US algorithm doesn't.

(c) What is the relationship between optimal planning in Blocksworld and the minimum hitting set problem? Explain in two to three sentences.

Exercise 3.2 (4 marks)

Consider the two STRIPS planning tasks $\Pi_1 = \langle V, I, O_1, \gamma \rangle$ and $\Pi_2 = \langle V, I, O_2, \gamma \rangle$ with $V = \{a_1, a_2, a_3, b_1, b_2, b_3, c, d\}$, $I(a_1) = \mathbf{T}$ and $I(v) = \mathbf{F}$ for all $v \in V$, $v \neq a_1$, $O_1 = \{o_1, o_2, o_3, o'_1, o'_2, o'_3, o'_4\}$, $O_2 = \{o_1, o_2, o_3, o''_1, o''_2, o''_3\}$, $\gamma = c$ and

$$o_1 = \langle a_1, a_2 \land \neg a_3 \land \neg c \rangle \qquad o'_1 = \langle d, b_1 \rangle \qquad o''_1 = \langle \top, \neg a_1 \land b_1 \rangle$$
$$o_2 = \langle a_1 \land a_2, a_3 \land \neg c \rangle \qquad o'_2 = \langle d, b_2 \rangle \qquad o''_2 = \langle \top, \neg a_1 \land b_2 \rangle$$
$$o_3 = \langle a_1 \land a_2 \land a_3, c \rangle \qquad o'_3 = \langle d, b_3 \rangle \qquad o''_3 = \langle \top, \neg a_1 \land b_3 \rangle$$
$$o'_4 = \langle b_1 \land b_2 \land b_3 \land d, c \rangle$$

Solve one of these tasks with breadth-first search with progression and the other with breadth-first search with STRIPS regression. In both cases, provide the breadth-first search tree and prune nodes that are duplicates of nodes that were generated before. In regression, additionally prune nodes with formula $\bot$. Depict nodes that are pruned in your search tree but mark them as pruned and do not expand them. You may combine multiple nodes representing identical states or formulas into one if they are generated from the same search node (label the arc of the search tree with all operator names in this case). You may stop the search when the goal node (progression) or a node containing the initial state (regression) is generated.
Hint: One of the tasks can easily be solved with breadth-first search with progression but requires many expansions with breadth-first search with regression, and vice versa for the other task. Determine which task can easily be solved in which search direction before you actually perform the search.

Exercise 3.3 (3 marks)

Consider the formula $\varphi = a \wedge (b \vee c)$ and the following operators:

- $o_1 = \langle \top, \neg a \rhd b \rangle$
- $o_2 = \langle d, a \wedge (e \rhd \neg b) \rangle$
- $o_3 = \langle \neg a, b \rangle$

Compute $regr(\varphi, o_1)$, $regr(\varphi, o_2)$ and $regr(\varphi, o_3)$. In all cases, simplify the result as much as possible. Provide all intermediate steps for the computation of $regr(\varphi, o_1)$. For $regr(\varphi, o_2)$ and $regr(\varphi, o_3)$, the final result is sufficient. You might receive partial points for a wrong result if intermediate steps are provided though.

Submission rules:

- Exercise sheets must be submitted in groups of three 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. After creating your zip file and before submitting it, open the file and verify that it complies with these requirements.
- 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.