Planning and Optimization

M. Helmert, G. Röger
C. Büchner, R. Christen, S. Dold

University of Basel
Fall Semester 2022

Exercise Sheet 3

Due: October 16, 2023

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 (1.5+1.5 marks)

Consider the propositional planning task $\Pi = \langle V, I, O, \gamma \rangle$ with

$$\begin{aligned} V &= \{a, b, c, d\} \\ I(v) &= \mathbf{F} \quad \text{for all } v \in V \\ O &= \{o_1, o_2, o_3, o_4\} \\ \gamma &= d \end{aligned}$$

and

$$\begin{aligned} o_1 &= \langle \neg b, c \rangle \\ o_2 &= \langle \top, b \rangle \\ o_3 &= \langle \neg a, a \rangle \\ o_4 &= \langle a, (b \rhd d) \wedge \neg c \rangle \end{aligned}$$

(a) Plot the search tree explored by a progression breadth-first search of Π. Generate successors in the order of increasing operator indices (i.e., o_1 before o_2 etc.) and break ties in the expansion order in favor of lower operator indices of achievers. Mark nodes that are duplicates of previously generated states as such and prune them (i.e., do not expand them further).

(b) Plot the search tree explored by a regression breadth-first search of Π. Generate and expand nodes in the order described in part (a). Simplify the state formula as much as possible at every node of the search tree. Mark nodes with an unsatisfiable formula or one that logically entails the state formula of a previously expanded node and prune them.

Exercise 3.2 (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 $\mathit{regr}(\varphi, o_1)$, $\mathit{regr}(\varphi, o_2)$ and $\mathit{regr}(\varphi, o_3)$. In all cases, simplify the result as much as possible. Provide all intermediate steps for the computation of $\mathit{regr}(\varphi, o_1)$. For $\mathit{regr}(\varphi, o_2)$ and $\mathit{regr}(\varphi, o_3)$, the final result is sufficient. You might receive partial points for a wrong result if intermediate steps are provided though.

Exercise 3.3 (1+1+1+1 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 questions. Use your own words in your answers to avoid plagiarism.

Consider the following paper to answer the questions below:

> Suda, Martin (2016). Duality in STRIPS Planning.
> In *Proceedings of the 8th Workshop on Heuristics and Search for Domain-Independent Planning (HSDIP 2016)*, pp. 21-27.

(a) Identify two formal differences between the definition of STRIPS planning tasks in the paper (Section 2) and the definition used in the lecture (Chapter B5, slide 22).

(b) Consider the following STRIPS planning task (according to the notation used in the paper): $\mathcal{P} = (X, I, G, A)$ where

- $X = \{w, x, y, z\}$;
- $I = \{x\}$;
- $G = \{y, z\}$; and
- $A = \{a_1, a_2\}$ with
  - $a_1 = (\{x\}, \{w, y\}, \{x, z\})$ and
  - $a_2 = (\{y\}, \{x, z\}, \{w\})$.

Provide the dual $\mathcal{P}^d$ of $\mathcal{P}$. Define all components explicitly (e.g., $\{w, x\}$ rather than $X \setminus G$).

(c) Summarize in 2–3 sentences the high-level proof idea of Theorem 3. Discuss in particular why progression and regression search are relevant.

(d) Describe in your own words and no more than 3 sentences: What makes an action *relevant* according to this paper and how is relevance exploited in regression search? What is its dual counterpart in progression search?

Submission rules:

- Exercise sheets must be submitted in groups of 2–3 students. Create a team on ADAM including all members of your group and submit a single copy of the exercises per group.
- 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 text file(s) 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 text file(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.