

Planning and Optimization

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

University of Basel
Fall Semester 2022

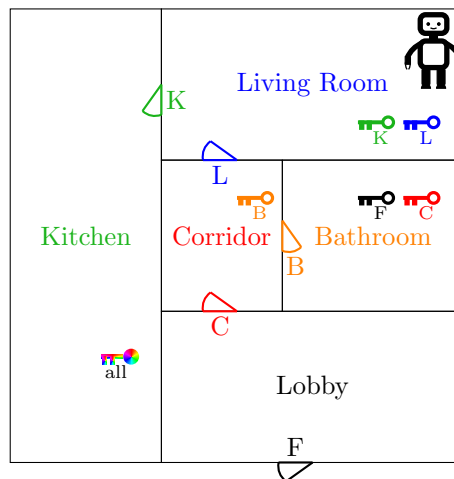
Exercise Sheet 2

Due: October 10, 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 2.1 (3+0.5+0.5 marks)

A bug in your household robot's software made it lock all doors inside and to your apartment and drop all keys in arbitrary positions. The robot has been updated and the bug is fixed, but you are now locked out of your apartment. Only the robot can let you back in by opening the front door. The current situation in your home is as follows:



The multi-colored key in the kitchen is the central key that locks and unlocks all doors, and the robot's current location is the living room. Keys and doors match based on the used color and annotation, e.g., the blue door between the living room and the corridor that is annotated with an 'L' can be opened with the blue key in the living room that is also annotated with an 'L'. Initially, all doors in the apartment are locked.

- (a) You can only control your robot by giving it a valid planning task II. Model II with actions
- to open a door (one for each fitting pair of door and key),
 - to pick up a key that is in the same room as the robot (one for each key)
 - and to move from one room to another through an open door that connects both rooms (one for each connected pair of rooms).

Assume that all actions incur a cost of 1, that the initial state is as described and depicted above and that the goal is that the front door 'F' is unlocked.

- (b) Provide an optimal plan for II. What is its cost?
- (c) Provide an optimal plan and its cost if only moving incurs a cost of 1 (and the cost of opening a door and picking up a key is 0).

Exercise 2.2 (0.5+0.5+1 marks)

Let $s = \{a \mapsto \mathbf{T}, b \mapsto \mathbf{F}, c \mapsto \mathbf{T}, d \mapsto \mathbf{T}\}$ be a state and o_1, o_2 be operators for a planning task Π , with

$$o_1 = \langle a \wedge \neg b, \neg a \wedge \neg d \wedge ((b \vee c) \triangleright a) \rangle \text{ and } o_2 = \langle b \wedge d, ((a \wedge b) \triangleright (c \wedge d)) \rangle.$$

- What is $\text{effcond}(\neg d, \text{eff}(o_1))$?
- What is $\text{effcond}(c, \text{eff}(o_2))$?
- Is o_1 applicable in s ? Is o_2 applicable in s ? If yes, provide $s[[o_1]]$ and/or $s[[o_2]]$.

It is sufficient to give the final results. In L^AT_EX, you can use `\top` for \top and `\bot` for \perp .

Exercise 2.3 (1+0.5+1+0.5+1 marks)

Let $\Pi = \langle V, I, O, \gamma \rangle$ be a planning task where

$$\begin{aligned} V &= \{a, b, c\} \\ I &= \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{T}\} \\ O &= \{o_1, o_2, o_3, o_4\} \text{ where} \\ &\quad o_1 = \langle a, \neg b, 1 \rangle \\ &\quad o_2 = \langle a \wedge b, \neg a \wedge c, 2 \rangle \\ &\quad o_3 = \langle \neg c, \neg a \wedge (b \triangleright a), 1 \rangle \\ &\quad o_4 = \langle \neg a \wedge c, (b \triangleright \neg c), 3 \rangle \\ \gamma &= \neg b \wedge \neg c. \end{aligned}$$

Construct step by step the transition system $\mathcal{T}(\Pi) = \langle S, L, c, T, s_0, S_* \rangle$ induced by the task Π .

- Define the set of states S of $\mathcal{T}(\Pi)$. The name of a state s should clearly express the value of all variables in s .
- Define the set of labels L and the cost function $c: L \mapsto \mathbb{R}_0^+$.
- Define the transition relation T .
- Define the initial state s_0 and the set of goal states S_* .
- Provide a graphical representation of $\mathcal{T}(\Pi)$.

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.