Introduction to Artificial Intelligence

Dr. Thomas Keller — University of Zürich
C. Büchner, C. Grundke, A. Kauffmann — Spring Semester 2022

Exercise Sheet 4

Due: March 23, 2022
Points total: 20 marks

Exercise 4.1 – Formalization **4 marks**

Formalize the following Latin square as a constraint satisfaction problem (CSP).

	1	2	3
A	1		3
B	2		
C			

In a Latin square, each symbol appears exactly once in every row and in every column. In the above Latin square, each row and each column must contain each number from 1 to 3 exactly once.

Exercise 4.2 – Consistencies **5 marks**

Consider the CSP $\mathcal{C} = \langle X, D, C \rangle$ with

- variables $X = \{x, y, z\}$;
- domains $D = \{\mathrm{dom}(x), \mathrm{dom}(y), \mathrm{dom}(z)\}$ such that

$$\mathrm{dom}(x) = \{2, 4, 6\}, \qquad \mathrm{dom}(y) = \{1, 4, 9\}, \qquad \mathrm{dom}(z) = \{0, 1, 2, 3\};$$

- and constraints $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ such that

$$c_1 = \langle (x), x \in \{2x' \mid x' \in \mathbb{N}\} \rangle, \qquad c_4 = \langle (x, y), x^2 = 4y \rangle,$$
$$c_2 = \langle (y), y \neq 3 \rangle, \qquad c_5 = \langle (y, z), y = z^2 \rangle,$$
$$c_3 = \langle (z), z < 4 \rangle, \qquad c_6 = \langle (x, z), x = 2z \rangle.$$

Answer the following questions. Justify your answer by explaining the meaning of the property in your own words if your answer is "yes", or by giving a counterexample if your answer is "no".

Is the constraint graph corresponging to $\mathcal{C}$...

(a) node-consistent? (1 mark)

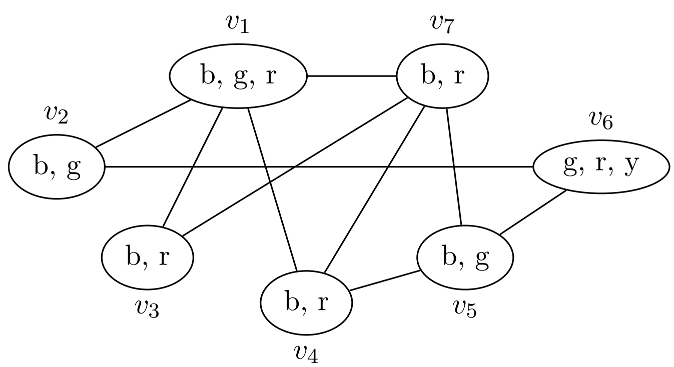
(b) arc-consistent? (1 mark)

(c) path-consistent? (2 marks)

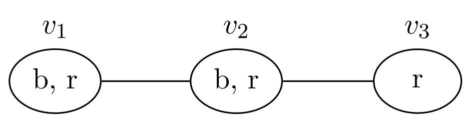
(d) strongly 3-consistent? (1 mark)

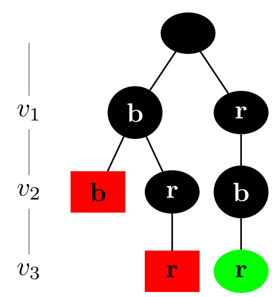
Exercise 4.3 – Backtracking-Search **7 marks**

Consider the constraint graph for a graph coloring problem as illustrated below. Each node represents one variable and the colors (i.e., b="blue", g="green", r="red", and y="yellow") annotated to each node represent its domain. A valid solution to the CSP assigns different colors to each pair of connected nodes.



(a) Provide the search tree that is created by applying backtracking-search on the depicted problem. Choose variables in an ascending order ($v_1 < v_2 < \cdots < v_7$) and values alphabetically (b < g < r < y). Use backtracking without inference (i.e., back up one variable whenever failure is detected). Depict the search tree along the lines of the following example:





(4 marks)

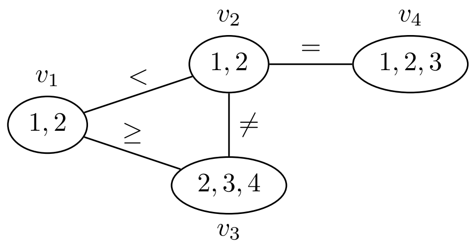
(b) Provide the variable and value ordering using the *minimum remaining values* and *least constraining value* heuristics. Would the resulting search be slower or faster than the one from part (a) of this exercise? Justify your answer. (2 marks)

(c) Explain why it is a good heuristic to choose the variable that is *most* constrained but the value that is *least* constraining in a CSP search. (1 mark)

Exercise 4.4 – AC3 with Forward Checking

4 marks

Consider the following (inconsistent) constraint graph $\mathcal{C}$:



(a) Provide the search tree when applying backtracking *with forward checking* using the minimum remaining values variable ordering, breaking ties by most constraining variable. Annotate each node in the search tree with the new domain for all variables whose domain changed.

Hint: We do not specify a value ordering here since $\mathcal{C}$ is inconsistent. You may choose any value ordering. (2 marks)

(b) Apply the AC-3 algorithm. Select the variables X_i and X_j in each iteration of the while loop such that the domain of X_i changes in the call to $\texttt{revise}(\mathcal{C}, X_i, X_j)$. Provide X_i, X_j, and $\text{dom}(X_i)$ in each iteration. Note that you do *not* have to provide the elements that are inserted into the queue, and you may stop the algorithm as soon as the domain of some variable is empty (which shows that the network is inconsistent). (2 marks)

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. Make sure your PDF has size A4 (fits the page size if printed on A4). Submit your single PDF file to the corresponding exercise assignment in MOODLE.
- For programming exercises, only create those code text files 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. Create a ZIP file (ending `.zip`, `.tar.gz`, or `.tgz`; *not* `.rar` or anything else) containing the code text file(s) (ending `.py`) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly.
- Do not upload several versions to MOODLE, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.