**Foundations of Artificial Intelligence**

M. Helmert
S. Eriksson
Spring Term 2021

University of Basel
Computer Science

# Exercise Sheet 7

**Due: April 21, 2021**

**Exercise 7.1** (2+1 marks)

(a) Formalize the propositional formula $\varphi = (x \vee \neg y) \wedge (\neg x \vee \neg z) \wedge (\neg x \vee z) \wedge (y \vee z)$ as a binary constraint network.

(b) Enumerate all solutions to your constraint network from (a).

**Exercise 7.2** (2 marks)

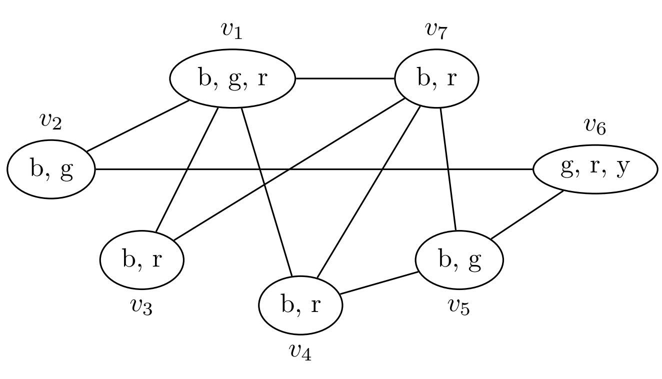
Consider the constraint network $\mathcal{C} = \langle V, \text{dom}, (R_{uv}) \rangle$ with

- $V = \{a, b, c\}$
- $\text{dom}(x) = \{0, 1, 2, 3\}$ for all $x \in V$
- $R_{a,b} = \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle\}$
  $R_{a,c} = \{\langle 0,3\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 3,0\rangle\}$
  $R_{b,c} = \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle\}$

Define the tightest network $\mathcal{C}'$ that is equivalent to $\mathcal{C}$. in other words find the unique $\mathcal{C}'$ such that $\mathcal{C}' \equiv \mathcal{C}$ and there is no $\mathcal{C}''$ with $\mathcal{C}'' \sqsubset \mathcal{C}'$ and $\mathcal{C}'' \equiv \mathcal{C}'$. Justify your solution.

**Exercise 7.3** (2+1 marks)

Consider the constraint network for the graph coloring problem that has been introduced on slide 10 of chapter 24 in the print version of the lecture slides:

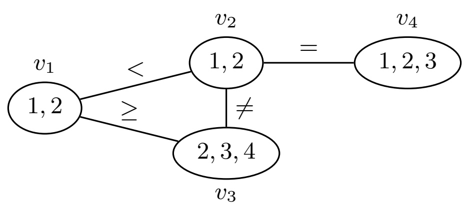



(a) Provide the search tree that is created by applying naive backtracking on the depicted problem. Choose variables in an ascending order ($v_1 < v_2 < \cdots < v_7$) and values alphabetically. Depict the search tree in a similar style as in the lecture slides.

(b) Was the chosen variable order good? Base your answer on a comparison of the size of your search tree to the one from the lecture (chapter 24, slide 9 handout version).

**Exercise 7.4** (1+1 marks)

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




(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.*

(b) Apply the AC-3 algorithm that has been presented on slide 21 of chapter 25 in the print version of the lecture slides on $\mathcal{C}$. Select the variables $u$ and $v$ in each iteration of the while loop such that the domain of $u$ changes in the call to $\texttt{revise}(\mathcal{C}, u, v)$. Provide $u$, $v$, and $\text{dom}(u)$ 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).

**Submission rules:**

Upload a single PDF file (ending .pdf). If you want to submit handwritten parts, include their scans in the single PDF. Put the names of all group members on top of the first page. Use page numbers or put your names on each page. Make sure your PDF has size A4 (fits the page size if printed on A4).