Foundations of Artificial Intelligence

T. Keller, F. Pommerening
S. Sievers
Spring Term 2023

University of Basel
Computer Science

Exercise Sheet 7

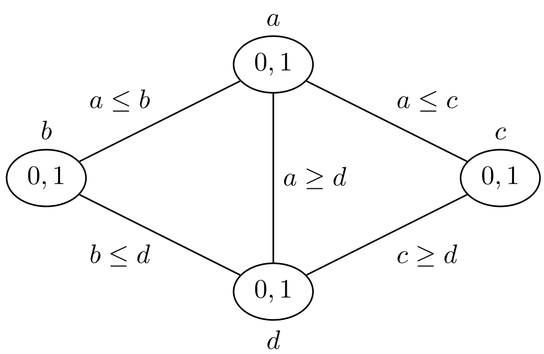
Due: April 23, 2023

Important: for submission, consult the rules at the end of the exercise. Non-adherence to the rules will lead to your submission not being corrected.

Exercise 7.1 (2 marks)

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

- $V = \{a, b, c, d\}$
- $\mathrm{dom}(x) = \{0, 1\}$ for all $x \in V$
- $R_{ab} = R_{ac} = R_{bd} = \{(0,0), (0,1), (1,1)\}$
  $R_{ad} = R_{cd} = \{(0,0), (1,0), (1,1)\}$



Apply the algorithm PC-2 (chapter 26) on $\mathcal{C}$ to compute an equivalent path consistent network. Select the variables u, v, and w in each iteration of the while loop such that R_{uv} changes in the call to `revise-3`($\mathcal{C}, u, v, w$). For each iteration, specify the call to `revise-3` and the updated constraint R_{uv}. 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 there are no u, v and w such that R_{uv} changes. Specify the full constraint network you obtained.

Exercise 7.2 (3 marks)

Let $\mathcal{C}$ be a solvable constraint network with an acyclic constraint graph. Show that the application of the algorithm for trees as constraint graphs (chapter 27 of the lecture slides) leads to a solution for $\mathcal{C}$ without needing to backtrack.

Hint: First use structural induction to show that after step 3 the domain of every node v contains only values for which a consistent assignment to all variables in the subtree rooted at v can be found. Then use this to show that backtracking with forward checking will always pick a value that can be extended to a solution. Also consider that finding a solution requires that no domain ever becomes empty.

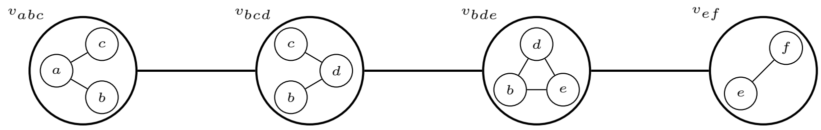
Exercise 7.3 (1+1.5 marks)

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

- $V = \{a, b, c, d, e, f\}$,

- $\mathrm{dom}_x = \{0, 1, 2, 3\}$ for all $x \in V$,
- $R_{a,b} = \{\langle x, y\rangle \mid x < y\}$,
  $R_{a,c} = \{\langle x, y\rangle \mid x = y - 1\}$,
  $R_{b,d} = \{\langle x, y\rangle \mid x > y\}$,
  $R_{b,e} = \{\langle x, y\rangle \mid x = y\}$,
  $R_{c,d} = \{\langle x, y\rangle \mid x = y\}$,
  $R_{d,e} = \{\langle x, y\rangle \mid x < y\}$,
  $R_{e,f} = \{\langle x, y\rangle \mid x < y\}$

In this exercise, you have to solve $\mathcal{C}$ with the tree decomposition approach presented in chapter 28 of the lecture slides. To do so, use the following tree decomposition $\mathcal{T}$:



(a) Provide the meta constraint network $\mathcal{C}^{\mathcal{T}}$ based on tree decomposition $\mathcal{T}$, using the variable names shown in $\mathcal{T}$.

(b) Solve the meta constraint network $\mathcal{C}^{\mathcal{T}}$ using the algorithm for trees (chapter 27 of the lecture slides). Use v_{abc} of $\mathcal{T}$ as the root node. Provide the resulting variable order, the list of calls to `revise` and the resulting updated domains, and the backtracking-free execution of `BacktrackingWithInference` with forward checking by specifying for each step which value you select for which variable and how the domains are updated through forward checking. Also specify the found solution to the original constraint network $\mathcal{C}$.

Exercise 7.4 (1+0.5 marks)
Consider the formula $\varphi = ((A \lor B) \to C) \land \lnot((C \lor B) \to A)$ and interpretation $I = \{A \mapsto \mathbf{F}, B \mapsto \mathbf{T}, C \mapsto \mathbf{T}\}$

(a) Show that $I \models \varphi$ using the definition of $\models$.

(b) Is φ falsifiable or valid? Justify your answer by providing an interpretation under which φ does not hold or explaining why not such interpretation exists.

Exercise 7.5 (1 marks)
Convert the formula $(\lnot P \lor Q) \to R$ to CNF by applying the logical equivalences from chapter 29 of the lecture.
Provide all intermediate formulas that result from applying an equivalence transformation.

Submission rules:

- Exercise sheets must be submitted in groups of two 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.
- 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.