SAARLAND UNIVERSITY — ARTIFICIAL INTELLIGENCE
Prof. Dr. Jörg Hoffmann, Dr. Daniel Fišer, Daniel Höller, Dr. Sophia Saller

Theoretical Exercise Sheet 5
Deadline Friday, May 20, 23:59

About the submission of this sheet.
- You might submit the solutions to exercises in groups of up to 3 students.
- All students of a group need to be in the same tutorial.
- Write the names of **all** students of your group on your solution.
- Hand in the solution **in CMS** and use "Team Groupings".
  - Go to your personal page in CMS. Here you find the entry "Teams".
  - When you click "Create team", you get an invite code.
  - Please share this with your team mates, who need to click on "Join team" and enter the code.

1. (11 points) (3+5+3) Your country has given you an important task. Multiple types of monsters are supposed to be spread around a piece of land. The piece of land is separated into different areas with different biomes. Below, you can see a map of the piece of land you are considering. The biomes are **swamp** (Green), **city** (Grey), and **countryside** (Brown). The monsters have different types, namely **water** (W), **fire** (F), **grass** (G), **ice** (I), **steel** (S), **rock** (R) and **electric** (E). Each monster type is supposed to be assigned to one area.

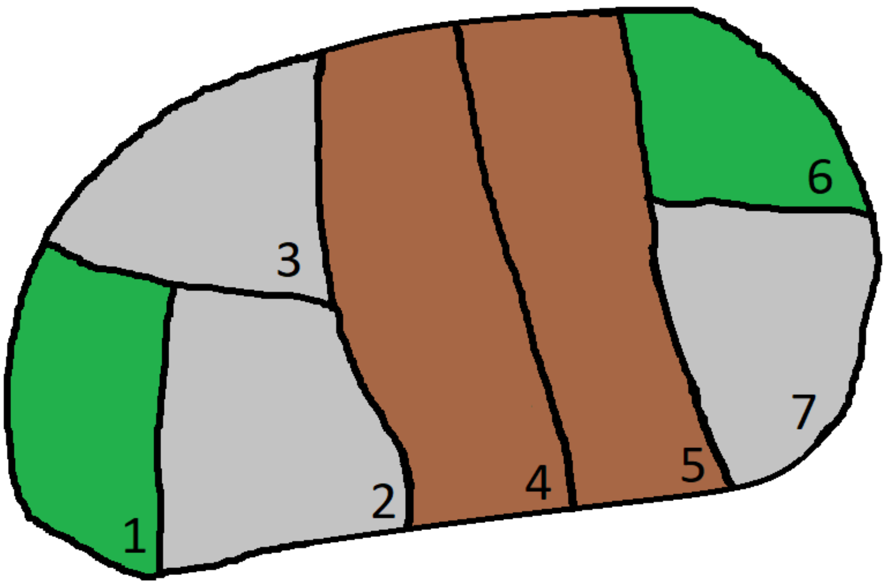
   The different types of monsters can live in specific biomes:

   - Water and rock monsters can live in all biomes.
   - Fire monsters can only live in the countryside.
   - Grass monsters can live in the swamp and in the countryside.
   - Ice, steel and electric monsters can only live in the city.

   The grass and water type monsters are picky when it comes to their neighbors. The water type monsters don't want fire, grass and electric type monsters as neighbors, while the grass type monsters don't want fire, ice and rock type monsters next to them.

   Sadly, team spaceship is in area 7, and they want to steal electric type monsters. They can reach area 7 and the adjacent areas.

   Your task is to formulate a CSP, which describes all the given details. Use the set of variables $V = \{v_1, ..., v_7\}$, where variable v_i models which type of monster can live in area i. This will be done in three steps.



1. Begin by describing the Domains for all areas, using the information what type can live in which biome. This means, give the domain D_{v_i} for each variable $v_i \in V$. Some areas have the same domain, as they have the same biome, you can make use of this.
2. Now, give the binary constraints $C_{v_i v_j} \subseteq D_{v_i} \times D_{v_j}$ for all relevant pairs of variables. This means, that for each two adjacent areas, give the pair of monsters, that can be in those areas, so that no neighbor preference of the water and grass monsters gets broken.
3. Lastly, give the unary constraints $C_{v_i} \subseteq D_{v_i}$ for the areas that are affected by team spaceship.

2. (9 points) (8+1) Now, you are tasked with organizing a hair dying conference. This year, the colors **red**, **green** and **blue** are trending, and every company at the conference wants to show their products. Each company gets one area to show their products, and to create difference between the areas, each company can only show one color, and no two adjacent areas can have the same color.

   Your task is to find a hair color for each area, so that **no two adjacent areas have the same colors**. A map of the conference hall, together with the areas (and their respective letter) is given below.

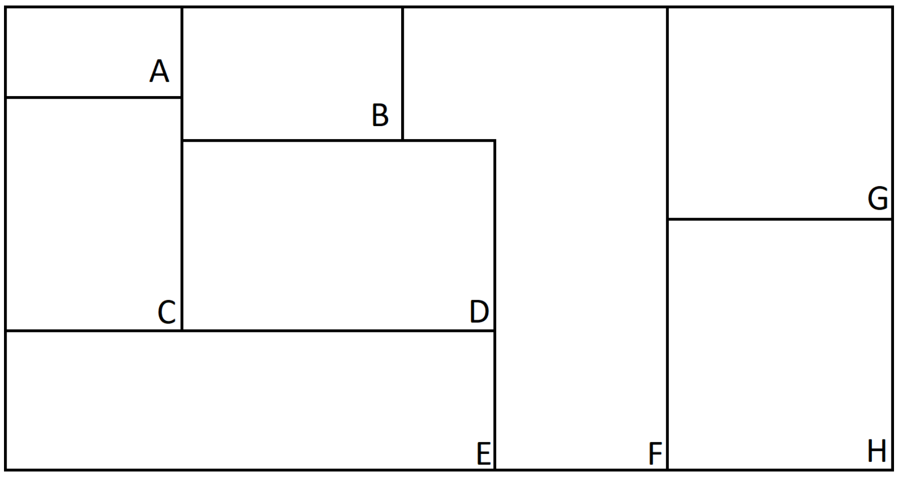
   To find a solution, use the naïve backtracking algorithm from the lecture. The constraint satisfaction problem is defined as:

   - The variables $V = \{A, B, C, D, E, F, G, H\}$, each with the domain $D = \{R, G, B\}$, where R stands for red, G for green and B for blue.

- The constraints are for all adjacent areas, to not have the same colors, so $C_{u,v} = D_u \times D_v \setminus \{(R,R),(G,G),(B,B)\}$ for all $u, v \in V$, where u and v are adjacent.

As variable ordering, pick among the **most constrained** variables the **most constraining**. As tie-breaker, use the **alphabetical order**, so A before B and so on. For each variable, **first try assigning R, then G and lastly B**.

As solution, hand in the search graph, **including the inconsistent partial assignments** you find when running the algorithm. Is your found solution the only one? If yes, justify your answer, if not, state a different solution (you don't have to run the algorithm again).



3. (4 points) Consider the following two constraint networks $\gamma = (V, D, C)$ and $\gamma' = (V, D, C')$ with the same set of variables and variable domains as shown in Figure 3:

   - V = {a,b,c,d}
   - D_v = {1,2,3} where $v \in V$

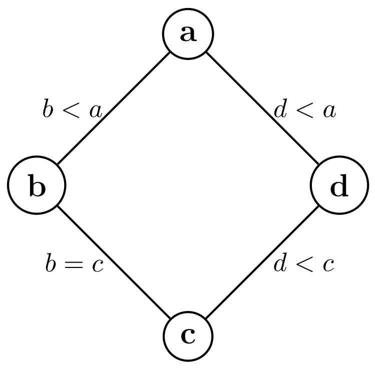


Figure 1: γ

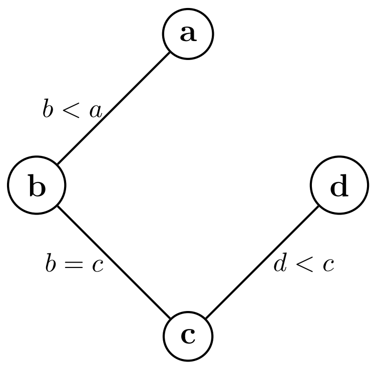


Figure 2: γ'

Mark all the statements that are True:

- ◯ a. γ and γ' are equivalent.
- ◯ b. γ is strictly tighter than γ'.
- ◯ c. γ is tighter than γ'.
- ◯ d. γ is solvable.

4. (20 points) (1 + 11 + (1+1+3+3)) Consider the following constraint network $\gamma = (V, D, C)$:

   - Variables: $V = \{a, b, c, d, e, f, g\}$
   - Domains: $\forall v \in V : D_v = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
   - Constraints:
     1. $|a^2 - b| < 15$
     2. $a \leq c$
     3. $b = c^2 - 2$
     4. $|d - c| \leq 4$
     5. $d < e - 6$
     6. $|3e - f^2| \leq 6$
     7. $|3g^2 - e| < 3$
     8. $f \cdot g = 12$

   a) Draw the constraint network γ.

   b) Run the AC-3 algorithm on γ as specified in the lecture. Specify the set M we have at the beginning. For each iteration of the while loop, give:

      - The pair (u, v) removed from M,
      - the domain of u after the call to $Revise(\gamma, u, v)$,

- whether the domain changed after the call,
- and the pairs (w, u) to be added into M.
- You **don't** need to add M for every iteration, only once at the beginning.

You can use a table like this:

Step	Removed (u,v)	New D_v	Changed	To be added
1				

NOTE 1: Initialize M as a lexicographically ordered list (that is, (a, b) would be before (a, c), and both before (b, a), and so on). Furthermore, use M as a FIFO queue; that is, always remove the element from the front and add new elements to the back.
NOTE 2: If any (u, v) pair that needs to be added at the end of each iteration is already present in M, **please underline them** to differentiate. For example, say $M = \{(x, y)\}$ and the pairs that need to be added are (m, n) and (x, y), then write $TBA = \{(m, n), \underline{(x, y)}\}$. If nothing needs to be added then write $TBA = \phi$.

c) Consider the same constraint network γ but **without the constraints 1 and 8**. Run AcyclicCG on γ as specified in the lecture.

- Pick 'a' as the root and draw the directed tree.
- Give the resulting variable order obtained by step 2 of the algorithm. Break ties by using the alphabetical order.
- List the calls to $Revise(\gamma, v_{parent(i)}, v_i)$ in the order executed by step 3, and for each of them give the resulting domain of $v_{parent(i)}$.
- For each recursive call to BacktrackingWithInference during step 4, give the domain D'_v of the selected variable v_i after Forward Checking, and give the value $d \in D'_{v_i}$ assigned to v_i. If there is a choice to select assignment from the domain, **choose the largest number first**.

5. (5 points) (1+1+3) Consider the following constraint network $\gamma = (V, D, C)$:

- Variables: $V = \{a, b, c, d, e, f, g, h, i\}$
- $\forall v \in V : D_v = \{1, 2, 3, 4\}$.
- Constraints: $a < b; a < d; b < c; c \neq d; c < e; d < e; d < f; f \leq e; f < g; i = f + 1; g = h + 2; h < f; i > h;$

a) Draw the constraint graph of γ.

b) What is an optimal (minimal) cutset V_o for γ?

c) If the CutsetConditioning algorithm from the lecture is called with such a minimal cutset V_o, how many calls to AcyclicCG will be performed in the worst case? Justify your answer.