Mini Project 1: Where there's smoke, there's fire!

Hand-in Instructions

- The assignment is due by February 18, 2026 at midnight.
- Please submit to Gradescope (accessible via Canvas). There will be **two** assignments on Gradescope which you will need to submit to: one for your written answers, and one for code upload.
- Your written answers should be in a single PDF file, ideally produced with LaTeX.
- The goal here is learning. You can work with other students, but be sure that, in the end, the solutions you submit were your work and that you understand them completely.
- You may use LLMs to help with this assignment; note, though, that we've tried these questions in several LLMs and they often produce incorrect answers, so the correctness is up to you!

1 Smoke and Fire

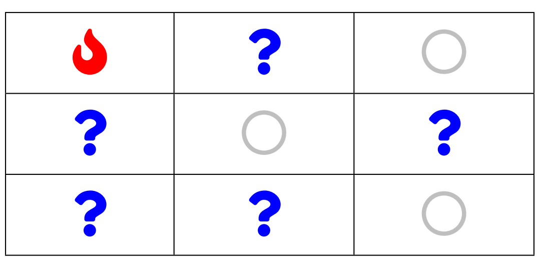
We will address an inference problem that is kind of like Minesweeper, by formulating it as a discrete constraint satisfaction problem (CSP) and solving it using a downloadable solver.

Our environment is a 2D rectangular grid. Every cell in the grid may have fire in it, may have smoke but no fire, or may be clear.

We will make some observations of the states of some cells in the grid. And we know some "physics" rules about how smoke and fire are geographically related (all within the same moment in time):

1. There is smoke at a location only if there is a fire in at at least one of the 4 adjacent locations.
2. If there is a fire at some location, then there is smoke *or* fire at every adjacent location.

Consider the following grid (circles indicate we have observed that cell to be clear):



Take a moment to run your human inference engine: which unknown values in the grid above can be determined?

1. (10 points) Write a Python program, using the `python_constraint` package, that can take in as input an arbitrary rectangular array with characters `'F'`, `'S'`, `'C'`, `'U'`, indicating fire, smoke, clear, and unknown, and produce a new array of characters of the same shape, in which any `'U'` character in a cell whose contents can be unambiguously determined from the input and the physics rules is replaced by a character indicating the inferred contents.

   What output does your code generate for the input below?

```
grid = [
['C', 'C', 'C', 'C', 'C', 'C', 'C'],
['C', 'C', 'S', 'U', 'S', 'C', 'C'],
['C', 'U', 'F', 'U', 'F', 'U', 'C'],
['C', 'U', 'U', 'U', 'S', 'C', 'C'],
['C', 'C', 'U', 'U', 'C', 'C', 'C'],
['C', 'U', 'C', 'U', 'U', 'C', 'C'],
['C', 'C', 'C', 'C', 'C', 'C', 'C']
]
```

2. (15 points) How did you decide whether to leave a cell as 'U' in your output?

3. (15 points) Draw the constraint graph for the 2 x 3 top cells in the example provided on page 1 ($[FUC; UCU]$). Remember the unary factors for the observations.

4. (10 points) Assume we start with a partial assignment of all the variables with given values determined. Consider the variable ordering of starting in the upper left-hand corner, filling out the top row from left to right, then the next row, etc. Is it possible to find a satisfying assignment to our first example problem on page 1, using forward checking, without backtracking, **for all value orderings?** If not, explain. If yes, what assignment is found?

   Note that The version of forward checking in the book assumes that we have binary constraints. You can use a variation on forward checking in which, when you make an assignment to a variable, you reduce the domains of any other variables that share a constraint as much as possible (but don't do any further propagation).

5. (10 points) If you had a large problem in which you knew that a whole row of cells across the center was clear, would that affect the worst-case complexity of solving an $n \times n$ problem by backtracking search? If so, how? Would the worst-case complexity class change?

2 Teaching Yourself

The example above was relatively simple to represent and solve. In contrast, most real-world problems can be modeled in several different ways, and there are many reasonable constraints one might impose on a model to obtain a solution. In practice, some constraints are more effective than others and can help us reach solutions much more quickly. Moreover, certain constraints allow more flexibility than others. In this problem, you will construct a model and describe the constraints used to represent the process of matching students with tutors.

A bit more formally, imagine we have a set T of tutors, set S of students, set H of hours in the week, and set C of classes. Each tutor $t \in T$ is qualified to teach a subset of classes $C_t \subseteq C$ and is available during a subset of hours $H_t \subseteq H$. Each student $s \in S$ requires help in a subset of classes $C_s \subseteq C$ and is available to receive tutoring during a subset of hours $H_s \subseteq H$.

6. (15 points) Your goal is to find a valid assignment of tutors to students. At any hour $h \in H$, a student may be assigned to at most one tutor for at most one class, and a tutor may teach at most one student for at most one class. Assignments must respect both the tutor qualifications and the availability constraints of tutors and students.

   Define a constraint satisfaction problem that models this real-world situation. Specify the variables, the domains of the variables, and the constraints in some formal notation (math or pseudocode).

7. (15 points) The model above can lead to many solutions that wouldn't be practical in the real world. What are some other constraints that you might want to add to this model to make it more realistic and increase the utility of the resulting solutions? List 3 considerations that this model does not account for, and how the model could be modified and constraints added to account for them. For each one,

(1) explain in English the weakness of the current formulation that you are addressing, (2) explain in English the constraint that you propose to add, and (3) provide a formal description of the constraint that's compatible with your formulation above.

8. (4 points) In our original model, is it possible that there could be more tutors than students, yet the demand still could not be met? Explain in 1-2 sentences.

9. (3 points) In our original model, could there exist satisfying assignments where no one is paired? Explain in 1-2 sentences.

10. (3 points) How would you deal with a situation where there are not enough tutors to meet the demand (think outside the box)? Explain in 1-2 sentences.