Artificial Intelligence 1 – WS 2020/2021
Assignment 4: Constraint Satisfaction
– Given Dec 3, Due Dec 13 –

Problem 4.1 (Scheduling CS Classes as a CSP) 40 pt

You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time. The classes are:

- Class 1 - *Intro to Artificial Intelligence*: meets 8:30-9:30am,
- Class 2 - *Intro to Programming*: meets 8:00-9:00am,
- Class 3 - *Natural Language Processing*: meets 9:00-10:00am,
- Class 4 - *Machine Learning*: meets 9:30-10:30am,
- Class 5 - *Computer Vision*: meets 9:00-10:00am.

The professors are:

- Professor A, who is available to teach Classes 1, 2, 3, 4, 5.
- Professor B, who is available to teach Classes 3 and 4.
- Professor C, who is available to teach Classes 2, 3, 4, and 5.

1. Formulate this problem as a binary CSP problem in which there is one variable per class, stating the domains, and constraints. Constraints should be specified formally and precisely.

2. Give the constraint graph associated with your CSP.

3. Give examples of
   - a total inconsistent assignment
   - a solution

Problem 4.2 (Finite CSPs in Prolog) 40 pt

In this exercise, we implement the CSP search algorithm and its heuristics in Prolog. We specialize to binary CSPs with finite domains.

The implementation is quite complex, and a large part of it is provided below. The concrete problem statement and some hints are embedded in the code as comments.

The code is available at `https://swish.swi-prolog.org/p/CSP_problem.swinb`

Because the code is so complex, there may be some bugs or ambiguities that we discover once you start working on it. In that case, we will update the notebook. (You can check the history on the server to see changes.)

Problem 4.3 20 pt

Assume a CSP with a ternary constraint

$$(x_1, x_2, x_3) \in C \subseteq D_1 \times D_2 \times D_3.$$

Show how this constraint can be replaced with binary constraints by introducing additional variables over appropriate domains.

Hint: You do not have to write a lot here. But you have to be explicit about how you change the variables, domains, and constraints.

Problem 4.4 (Kalah Tournament) 100 pt

This is an extraordinary problem, in which we implement adversarial search as a tournament. You can implement all search methods, e.g., to simulate a move or compute the full game tree etc.

Submission parameters:

- Team size: as before, 3 people per team
- Deadline: 2021-01-06
- Site: The submission site will be opened later.
- Format: Submissions must conform to the Kalah framework at
  `https://github.com/KWARC/Kalah-Framework`
- Points: The team with the best agent receives an additional(!) 100 points, the 2nd team 90 points, the 3rd 80 etc.

Competition rules:

- We use the Kalah rules as described at the introduction section of `https://en.wikipedia.org/wiki/Kalah` with variable numbers of houses and seeds
- Static resource constraint: Your submission must not exceed 200MB.
- Dynamic resource constraints: The time limit for moves will be in the range of seconds and memory limits will correspond to that. But your agent will not know exactly how much time and memory it has left for its move. Your agent should regularly update the variable holding the best move; if it runs outs of memory or time, the last update will be used.
- Disqualification: Any failure such as throwing an exception (other than for running out of resources) or making an illegal move may result in the loss of the game. An agent repeatedly throwing exceptions may be disqualified from the tournament.

Further details will be announced in the forum as they come up.