CSE 412A Spring 2022 — Introduction to Artificial Intelligence

Exercise 6

- You have approximately as many minutes as there are points.
- Mark your answers ON THE EXERCISE ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.
- For True/False questions, please *circle* your answer.

First name	
Last name	
WUSTL ID	

For staff use only:

Q1.	Constraint Satisfaction Problems	/20
	Total	/20

THIS PAGE IS INTENTIONALLY LEFT BLANK

True/False : 13
Search : 30
Adv. Search : 17
CSPs : 20

80

Variables : A1, N, ..., R

Values : (x, y) $\quad x \in \{AM, PM\}$, $\; y \in \{R1, \dots, R6\}$.

Const. : $(A1, N) \in \{((AM, x), (PM, y)), ((PM, x), (AM, y))\}$ where $x, y \in \{R1, \dots, R6\}$

A1's time $\neq$ N's time

Q1. [20 pts] Constraint Satisfaction Problems

You have been asked to help in room assignments for the following courses: Artificial Intelligence (AI), Robotics (R), Java (J), Systems (S), Philosophy (P) and Networks (N). You have the following information available:

- There are two time slots available for these courses: AM, PM.
- There are five rooms available for these courses: R1, R2, . . . , R5.
- The following pairs of courses have overlapping students: Courses AI and N, and courses S and P. These pairs of courses can not be held at the same time slot.
- The following pairs of courses are being taught by the same professor: Courses AI and R, and courses N and J. These pairs of courses can not be held at the same time slot.
- Finally, course R needs to be taught in the robotics lab (R1) and courses J and S need to be taught in a computer lab (R3).

Needless to say, no two courses can be held at the same time and in the same room.

(a) [15 pts] Formulate this problem as a CSP, clearly identifying the variables, their possible values, and the constraints.

Variables : AI, R, J, ..., N
Values : {(AM, R1), (AM, R2), ..., (PM, R5)}.

Constraints :

- AI ≠ R ≠ J ≠ ...
- (AI, N) ∈ {(AM, x), (PM, x)), ...

equiv
- (AI[0], N[0]) ∈ {(AM, PM), (PM, AM)}
- AI[0] ≠ N[0]

- (AI, N) ∈ {((AM, R1), (PM, R1)), ((AM, R2), (PM, R2)), ..., ((AM, R1), (PM, R2))
- R1 ∈ {(AM, R1), (PM, R1)}.
- alldiff (AI, R, J, ..., N)

Explicit : (A, B) ∈ {(R, G), (R, B), (G, B), ...}
Implicit : A ≠ B

A —≠— B
= {R, G, B} = {R, G, B}

Variables : AI.t, AI.l, R.t, R.l, ..., N.t, N.l
Values : {AM, PM} for AI.t, R.t, ..., N.t
{R1, ..., R5} for AI.l, R.l, ..., N.l

Constraints : AI.t ≠ N.t
S.t ≠ P.t
AI.t ≠ R.t
R.l = R1
J.l = S.l = R3
alldiff((AI.t, AI.l), (N.t, N.l), ...)

(b) Each question is worth 1 point. Leaving a question blank is worth 0 points. **Answering a question incorrectly is worth −1 point.** This gives you an expected value of 0 for random guessing.

(i) [1 pt] [*true* or *false*] The Minimum Remaining Value (MRV) heuristic is a variable-ordering heuristic.

(ii) [1 pt] [*true* or *false*] Breadth-first search always expands at least as many nodes as depth-first search when solving CSPs with the same variable- and value-ordering heuristics.

(iii) [1 pt] [*true* or *false*] Depth-first search always expands at least as many nodes as a search algorithm that chooses random fringe nodes to expand when solving CSPs with the same variable- and value-ordering heuristic.

(iv) [1 pt] [*true* or *false*] After running arc consistency on a CSP, it is impossible for backtracking to prune a value during forward checking.

(v) [1 pt] [*true* or *false*] Hill climbing with a random initial assignment is guaranteed to find a solution to a CSP, if one exists.