EECS 492: Introduction to AI
Homework 3 (100 pts)

General Information

Due: 11:59 PM, October 20, 2016
Notes:

- For the written questions, put your answers in a pdf (with your name in the pdf) and submit to the corresponding assignment on Gradescope.
- For the coding questions, submit your code file **firstname_lastname_solver.py** to the corresponding assignment on canvas, and also **append the code** to the end of your pdf.
- Late homework will be penalized 10% per day (each day starts at 11:59 PM on the due day).
- Homework turned in after **one** late day **will not be accepted**.

Note: **There will be only one day of late submissions for this assignment – do not leave it until the last minute!**

Problem 1 (10 pts)

1. State whether each statement is true or false.
   a. $(A \wedge B) \models (A \Leftrightarrow B)$
   b. $A \Leftrightarrow B \models \neg A \vee B$
   c. $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$
   d. $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$
   e. $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$
   f. $(A \vee B) \wedge \neg(A \Rightarrow B)$ is satisfiable

2. Prove each of the following assertions ($\alpha \models \beta$ iff in every model in which α is true, β is also true):
   a. α is valid if and only if True $\models \alpha$
   b. $\alpha \models \beta$ if and only if the sentence ($\alpha \Rightarrow \beta$) is valid
   c. $\alpha \equiv \beta$ if and only if the sentence ($\alpha \Leftrightarrow \beta$) is valid
   d. $\alpha \models \beta$ if and only if the sentence ($\alpha \wedge \neg \beta$) is unsatisfiable

Problem 2 (10 pts)

1. Convert the following knowledge base to conjunctive normal form (CNF)

- A
- $\neg C$

- $A \Rightarrow B$
- E
- $\neg C \Rightarrow D$
- $B \wedge D \wedge E \Rightarrow F$

2. Use resolution refutation to prove F

Problem 3 (10 pts)

Determine whether or not the following pairs can be unified. If they can be unified, present the **most general unifier** (MGU) and show the result. If they cannot be unified, explain why. Use the convention that variables are lowercase letters while constants are capital letters. Functions are represented as either capital letters or as "Has" or "Finds" or "GameOf". Use the unique-names assumption (Figure 9.1 in the text).

1. P(x, y, y)	P(A, B, C)
2. P(B, GameOf(x))	P(x,B)
3. C(B,FriendOf(B), z)	C(B, FriendOf(B), GameOf(FriendOf(B)))
4. P(x, y)	P(A, FriendOf(B))
5. Q(B, A)	R(x, FriendOf(y))
6. Q(FriendOf(x,x), y)	Q(FriendOf(A,C), B)

Problem 4 (15 pts)

Consider the following sentences in our knowledge base:

- Pat is a student.
- Ayse is a restaurant.
- Pat gets a scholarship.
- If any student gets a scholarship, it is a special occasion for that student.
- Pat is a patron of some restaurant.
- Whenever a student is a patron of a restaurant and it is a special occasion, then the student eats lentil soup.

Use the following conventions:

- Student(x) – x is a student.
- Restaurant(x) – x is a restaurant.
- GetsScholarship(x) – x gets a scholarship.
- SpecialOccasion(x) – it is a special occasion for x.
- Patron(x, y) – x is a patron of y.

- Eats(x, y) – x eats y.

Take the following steps:

1. Translate the knowledge base into first-order logic (use quantifiers!).
2. Convert to conjunctive normal form (CNF).
3. Prove that Pat eats lentil soup using resolution refutation.

Make sure to show all substitutions (e.g., x/Pat}.

Problem 5 (15 pts)

A project manager (PM) must schedule incremental deliverables for an upcoming video game release. There are seven different consecutive deliverable dates (Days 1 - 7) suggested by the higher ups. They are numbered one through seven in temporal order. Seven different deliverables – B, C, D, F, H, J, and K – must be assigned to the dates. Only one deliverable can occupy any date.

The assignment of the deliverables to the slots is subject to the following restrictions:

i. B and D must occupy consecutive dates (in either order).
ii. B must be delivered at an earlier date than K.
iii. D must be delivered at a later date than H.
iv. If H does not occupy the fourth date, then F must occupy the fourth date.
v. K and J cannot occupy consecutively numbered dates (in either order).
vi. Variables must be all different (Alldiff(B, C, D, F, H, J, K))

Part I:

In this part you will describe the constraint satisfaction problem presented above.

1. List the unary constraints, if any.
2. Which of constraints (i)-(v) are binary, if any? (Note that constraint (vi) could be thought of as a set of binary inequality constraints).

Part II:

In this part you will answer a series of questions about the logic problem presented above. **Make sure to show all work**.

1. Which of the following could be a possible ordered list of the deliverables? For sets that do not represent consistent assignments, list the constraints that are violated.

    - BDFHJCK
    - CJBHDKF

- HBDFJCK
- HDBFKJC
- HJDBFKC

2. Give the domains after applying arc consistency
   - B:
   - C:
   - D:
   - E:
   - F:
   - H:
   - J:
   - K:

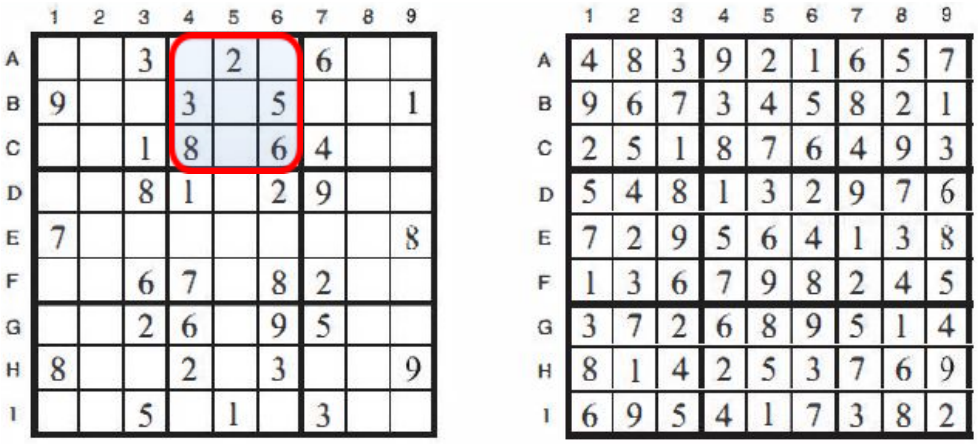
3. Perform backtracking search with forward-checking using the minimum remaining value heuristic (MRV), or alphabetical order if tied (see lecture 8 page 50 slide 99~100 for formatting), indicating backtracking moves. Add more rows to the table if necessary:

	B	C	D	F	H	J	K
initial	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7	1,2,3,4, 5,6,7

Problem 6 [40 pts]

You will solve a constraint satisfaction problem (a logic puzzle) both by hand and by implementing the AC-3 algorithm.

Consider the Sudoku puzzle as pictured below. Each variable is named by both its row and column. For example, the variable in the upper left corner is A1.



	1	2	3	4	5	6	7	8	9
A			3		2		6		
B	9			3		5			1
C			1	8		6	4		
D			8	1		2	9		
E	7								8
F			6	7		8	2		
G			2	6		9	5		
H	8			2		3			9
I			5		1		3		

	1	2	3	4	5	6	7	8	9
A	4	8	3	9	2	1	6	5	7
B	9	6	7	3	4	5	8	2	1
C	2	5	1	8	7	6	4	9	3
D	5	4	8	1	3	2	9	7	6
E	7	2	9	5	6	4	1	3	8
F	1	3	6	7	9	8	2	4	5
G	3	7	2	6	8	9	5	1	4
H	8	1	4	2	5	3	7	6	9
I	6	9	5	4	1	7	3	8	2

Each variable must be assigned a value from 1 to 9 subject to the constraint that no two cells in the same row/column/box may contain the same digit.

Part I: (10 pts)

1. List the variables in the upper center box (circled in red) and their corresponding initial domains.
2. Reduce the domain for variables in this box by enforcing the arc constraints with the entire puzzle. List the new domains. Are we forced into any assignments to these variables?
3. Let's assume that we had to choose one of the four unassigned variables in this box to explore further. If we were to employ the minimum remaining value heuristic, which variable (or variables) would we consider next?

Part II: (30 pts)

Use the provided starter code to implement the AC-3 algorithm.

Notice that the previous strategy is not enough to solve many Sudoku puzzles. We will address this challenge using backtracking search. Use or modify your AC-3 algorithm from Part I and implement backtracking search using the minimum remaining value heuristic. Test your code on the provided set of puzzles (sudokus.txt). Note that 0 in the sudoku file indicates an unassigned value. Replace 0 with the solution value once solved.

Your code will be run with the following command:

```
python firstname_lastname_solver.py <sudoku_filename> <type>
```

<type> can be either 'ac3' or 'bts', and <sudoku_filename> can be 'sudokus.txt'.

The program will generate a txt file (sudokus_solution_type.txt) that contains the solutions to the provided set of puzzles. Submit the generated txt files (for both ac3 and bts) with your code. We will test your code on a subset of provided sudokus.
10 pts for code and 5 pts for running, for each algorithm.