# CS 540: Introduction to Artificial Intelligence
# Homework Assignment #3: CSP and Logic

**Assigned: Friday, October 10**
**Due: Monday, October 20**

**Late Policy**:

Homework must be handed in by noon on the due date and electronically turned in by this same time.

- If it is 0 – 24 hours late, including weekend days, a deduction of 10% of the maximum score will be taken off in addition to any points taken off for incorrect answers.
- If it is 24 – 48 hours late, including weekend days, a deduction of 25% of the maximum score will be taken off in addition to any points taken off for incorrect answers.
- If it is 48 – 72 hours late, including weekend days, a deduction of 50% of the maximum score will be taken off in addition to any points taken off for incorrect answers.
- If it is more than 72 hours late, you will receive a '0' on the assignment.
- In addition, there are 2 'late days' you may use any time throughout the semester. Each late day has to be used as a whole – you can't use only 3 hours of it and "save" 21 hours for later use.

**Collaboration Policy**:

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TA, and instructor in order to help you answer the questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code or help on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

## Question 1: [10] Models

Consider a vocabulary with only four propositions (symbols), A, B, C, and D. How many models (i.e., interpretations in which all sentences are true) are there for the following set of three sentences? Show these models.

(a) $A \wedge B$
(b) $(A \wedge B) \vee (C \vee D)$
(c) $(B \Leftrightarrow C) \Leftrightarrow D$

## Question 2: [10] Soundness and Completeness

Abduction is an inference rule that infers P from $P \rightarrow Q$ and Q.

(a) Prove whether or not abduction is a **sound** rule of inference for Propositional Logic.

(b) Prove whether or not abduction is a **complete** rule of inference for Propositional Logic.

## Question 3: [10] Valid and Unsatisfiable Sentences in Propositional Logic

Is each of the following sentences in Propositional Logic **valid**, **unsatisfiable**, or **neither**? Justify your answer.

(a) $(\neg P \vee \neg Q) \Leftrightarrow (P \wedge Q)$
(b) $(P \Rightarrow \neg Q ) \Leftrightarrow (Q \Rightarrow \neg P)$
(c) $((P \Rightarrow Q) \wedge Q) \Rightarrow P)$

## Question 4: [15] Deductive Inference in Propositional Logic

Prove that implication is distributive in Propositional Logic by showing that

$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$ using

(a) The truth table (i.e., model checking) method.
(b) The resolution (i.e., resolution refutation) method, using the resolution rule of inference after converting to CNF.

## Question 5: [15] Translation to First-Order Logic

Problem 8.6, parts (a), (b), (c), and (d), and Problem 8.7 in the textbook. Use only the following predicates:

*Student*($x$),
*Takes*($x,y,z$) meaning student $x$ takes course $y$ in semester $z$,
*Passes*($x,y,z$) meaning student $x$ passes course $y$ in semester $z$,
$>(x,y)$ (or $x > y$) means $x$ is greater than $y$,
$=(x,y)$ (or $x = y$) means $x$ is equal to $y$,

*German*($x$),
*Language*($x$),
*Speaks*($x$,$l$) meaning person $x$ speaks language $l$.

The function *Score*($x$,$y$,$z$) means the score obtained by student $x$ in course $y$ in semester $z$.

## Question 6: Constraint Satisfaction

Consider the 6-Queens problem in which there is a 6 x 6 board and the goal is to place six queens (as used in Chess) at positions so that no two queens are in the same row, column, left diagonal, or right diagonal as any other queen. One way to formulate this as a constraint satisfaction problem is to have a variable $V_i$ for the queen positioned in column $i$, where the value specifies the row number where the queen is placed. Order the variables left to right. Order the values for each variable from bottom to top of the board.

(a) [10] Say we place the column-one queen in the 4th row, i.e., $V_1 = 4$, as shown:

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| Q | | | | | |
| | | | | | |
| | | | | | |

Show the result of the Forward Checking algorithm from this initial assignment by making a copy of the above initial board and adding an “X” in each square that is deleted from the corresponding column’s variable by this procedure.

(b) [10] Show the result of the Forward Checking algorithm after continuing from your result in (a) all the way down (depth-first) the first branch at each node in the search tree to assign the successive variables and to apply Forward Checking until either a solution is found or a backtrack point is reached. Show the resulting board with squares that contain either Q, X or are blank. Indicate whether this position represents a solution or a backtrack point.

(c) [10] Starting again with your result in (a), perform Forward Checking and also use the least-constraining value heuristic to pick the value to instantiate for each variable. In the case of ties, use the value closest to the bottom of the board. As in (b), continue down the search tree following the first branch at each step until either a solution is found or a backtrack point is reached. Show your result in the same way as (b). Indicate whether this position represents a solution or a backtrack point.

(d) [10] Same as (c) except start again with your result in (a) and use the Arc Consistency (AC-3) algorithm at each step of the search along the left-most branches in the search tree until either a solution is found or a backtrack point is reached. Show the resulting board and indicate whether this position represents a solution or a backtrack point.