INSTRUCTIONS

- **Due: Tuesday, 24 September 2019 at 10:00 PM EDT.** Remember that you have NO slip days for Written Homework, but you may turn it in up to 24 hours late with 50% Penalty.
- **Format:** Submit the answer sheet pdf containing your answers. You should solve the questions on this handout (either through a pdf annotator, or by printing, then scanning). Make sure that your answers (typed or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 15-281 and click on the submission titled HW4 and upload your pdf containing your answers.
- **Policy:** See the course website for homework policies and Academic Integrity.

Name	
Andrew ID	
Hours to complete?	

For staff use only

Q1	Q2	Q3	Total
/45	/25	/30	/100

Q1. [45 pts] Logic Warmup

(a) [6 pts] Which of the following entailment statements are true? Select all that apply.

☐ $(A \wedge B) \models (A \Leftrightarrow B)$

☐ $(A \Leftrightarrow B) \models (A \vee B)$

☐ $(A \Leftrightarrow B) \models (\neg A \vee B)$

☐ $((A \wedge B) \Rightarrow C) \models ((A \Rightarrow C) \vee (B \Rightarrow C))$

☐ $((A \vee B) \wedge (\neg C \vee \neg D \vee E)) \models (A \vee B)$

☐ $((A \vee B) \wedge (\neg C \vee \neg D \vee E)) \models ((A \vee B) \wedge (\neg D \vee E))$

☐ None of the above

(b) [6 pts] Out of the following statements, select all that are true.

☐ Any non-empty propositional clause is by itself satisfiable.

☐ Suppose a propositional clause contains 3 literals, each mentioning a different variable. The clause is satisfied in exactly 7 of the 8 possible models for the three variables.

☐ A conjunction of the following set of clauses is satisfiable.
$(A \vee B \vee C), (\neg A \vee B \vee C), (A \vee \neg B \vee C), (A \vee B \vee \neg C), (A \vee \neg B \vee \neg C), (\neg A \vee B \vee \neg C), (\neg A \vee \neg B \vee C), (\neg A \vee \neg B \vee \neg C)$

☐ None of the above

(c) [4 pts] We want to convert $A \Leftrightarrow (B \vee C \vee D)$ to conjunctive normal form. For each step, write down which rule is being used.

Step	**Rule/Explanation**
$A \Leftrightarrow (B \vee C \vee D)$	N/A
$(A \Rightarrow (B \vee C \vee D)) \wedge ((B \vee C \vee D) \Rightarrow A)$	
$(\neg A \vee (B \vee C \vee D)) \wedge (\neg(B \vee C \vee D) \vee A)$	
$(\neg A \vee B \vee C \vee D) \wedge ((\neg B \wedge \neg C \wedge \neg D) \vee A)$	
$(\neg A \vee B \vee C \vee D) \wedge (\neg B \vee A) \wedge (\neg C \vee A) \wedge (\neg D \vee A)$	

(d) [4 pts] Pacman has lost the meanings for the symbols in his knowledge base! Luckily he still has the list of sentences in the KB and the English description he used to create his KB. For each English sentence on the left, there is a corresponding logical sentence in the knowledge base on the right (not necessarily the one across from it). Your task is to recover this matching. Once you have, please fill in the blanks with the English sentence that matches each symbol.

English	**Knowledge Base**
There is a ghost at (0, 1).	$(C \vee B) \wedge (\neg C \vee \neg B)$
If Pacman is at (0, 1) and there is a ghost at (0, 1), then Pacman is not alive.	$C \wedge \neg D$
	$\neg A \vee \neg(B \wedge D)$
Pacman is at (0, 0) and there is no ghost at (0, 1).	D
Pacman is at (0, 0) or (0, 1), but not both.	

A =	**B** =
C =	**D** =

(e) [5 pts] Consider a world with 6 logical propositions: P, A, T, R, O, and X. How many different models of (P, A, T, R, O, X) satisfy each of the following knowledge bases?

Knowledge Base	# of Models
$P \vee A$	(i)
$T, O \wedge R$	(ii)
$(A \wedge \neg R) \vee (O \Leftrightarrow \neg P)$	(iv)
$A \wedge \neg R \wedge O \wedge \neg X$, $(O \Rightarrow X)$	(v)
$P \wedge A \wedge T$, $R \wedge O \wedge X$	(vi)

(f) [6 pts] The relationship in the Wumpus world between pits and breezes can be expressed as a biconditional. Write down the biconditional for $B_{2,2}$ and $P_{*,*}$ in a $4 * 4$ wumpus world then convert the biconditional to conjunctive normal form.

Biconditional:

Steps:

Conjunctive Normal Form:

For each of the following sentences in propositional logic, indicate whether it is *satisfiable* or *unsatisfiable*. If satisfiable, give a model such that the sentence is satisfied. Prove your answer by reducing the sentence to its simplest form. Remember to **show all the steps and write down an explanation of each step.** Let T stand for the atomic sentence *True* and F for the atomic sentence *False*.

(g) [7 pts] $(\neg(y \lor \neg y) \lor x) \land (x \lor (z \Leftrightarrow \neg z))$

Steps:

◯ Satisfiable ◯ Unsatisfiable

If satisfiable, give a model:

(h) [7 pts] $\neg(x \lor \neg(x \land (z \lor T))) \Rightarrow \neg(y \land (\neg y \lor (T \Rightarrow F)))$

Steps:

◯ Satisfiable ◯ Unsatisfiable

If satisfiable, give a model:

Q2. [25 pts] Resolution

(a) [20 pts] Given the following propositional logic clauses, show E must be true by adding $\neg E$ and using only the resolution inference rule to derive a contradiction. Your answer should be in the form of a graph, where each resolvent is connected by lines to its two parent clauses. Use the clauses below as the initial set of nodes in the graph.

Note: You do not need to use all the nodes, and you may use a node more than once.

$B \vee \neg C$ $A \vee \neg B$ $A \vee \neg D$ $E \vee \neg A \vee \neg B$ C $\neg D$

(b) [5 pts] Can we use forward-chaining on the question above? Please briefly explain why or why not.

○ Yes ○ No

Explain:

Q3. [30 pts] DPLL

The DPLL satisfiability algorithm is a backtracking search algorithm with 3 improvements: PURE-SYMBOLS, UNIT-CLAUSES, and EARLY TERMINATION. In this question, we'll ask you to relate these improvements (if possible) to more general CSP techniques used with backtracking search.

(a) [5 pts] The PURE-SYMBOL technique finds a propositional symbol that only occurs with the same "sign" throughout the expression and assigns it the corresponding value. Such symbols are called pure. For the following CNF expression and partial model, which symbols are pure and what value does this process assign to those symbols?

$$(C \lor D) \land (C \lor \neg A \lor D) \land (\neg D \lor A \lor E) \land (\neg B \lor A) \land (E \lor B \lor \neg D \lor \neg C)$$

$$\{C = \text{True}, E = \text{False}\}$$

Pure symbols:	**Values assigned:**

(b) [5 pts] Which of the following CSP techniques is most closely related to the PURE-SYMBOLS technique when applied to SAT for CNF sentences?

- ◯ MINIMUM-REMAINING-VALUES
- ◯ FORWARD-CHECKING
- ◯ LEAST-CONSTRAINING-VALUE
- ◯ BACKTRACKING
- ◯ No equivalent CSP technique

(c) [5 pts] The UNIT-CLAUSE technique finds all clauses that contain a single literal and assigns values to those literals. For the following CNF expression and partial model, which clauses are unit clauses?

$$(C \lor D) \land (\neg A) \land (\neg D \lor A \lor E) \land (B) \land (E \lor B \lor D \lor C)$$

$$\{D = \text{False}\}$$

Unit clauses:

(d) [5 pts] Which of the following CSP techniques is most closely related to the UNIT-CLAUSE technique when applied to SAT for CNF sentences?

- ◯ MINIMUM-REMAINING-VALUES
- ◯ FORWARD-CHECKING
- ◯ LEAST-CONSTRAINING-VALUE
- ◯ BACKTRACKING
- ◯ No equivalent CSP technique

(e) [5 pts] DPLL performs early termination in two steps: SUCCESS-DETECTION and FAILURE-DETECTION. First, SUCCESS-DETECTION checks to see if *all* clauses evaluate to true. If so, the sentence is satisfiable and any unassigned propositional symbols can be assigned arbitrarily. Next, FAILURE-DETECTION checks if *any* clause evaluates to false. In this case the sentence is unsatisfiable and this branch of the search can be pruned. How does this strategy relate to the early termination strategy used by the general BACKTRACKING algorithm?

◯ BACKTRACKING does both SUCCESS-DETECTION and FAILURE-DETECTION

◯ BACKTRACKING does only SUCCESS-DETECTION

◯ BACKTRACKING does only FAILURE-DETECTION

◯ BACKTRACKING does neither

(f) [5 pts] Assume you have an implementation of DPLL-SATISFIABLE(SENTENCE) that returns true if SENTENCE is satisfiable, false otherwise. Write the pseudocode for RESOLUTION(KB, QUERY-SENTENCE) that makes use of DPLL-SATISFIABLE.

RESOLUTION(KB, QUERY-SENTENCE)
input: KB **in CNF**
input: QUERY-SENTENCE **in propositional logic**
Returns true if the KB **entails the** QUERY-SENTENCE**, false otherwise**