CS267A: Probabilistic Programming and Relational Learning **Fall 2018**

Homework 1: Due Friday Oct. 12, 11:59PM

Instructions: Upload two files to CCLE: (1) a PDF typeset using LaTeX containing your solutions; (2) a zip file containing your code. Late work will not be accepted. See the syllabus for policies about collaboration and academic honesty.

Problem 1

Three people, A, B, and C, are suspected of a crime. They testify as follows:

- A says: B is guilty if C is innocent.
- B says: If A is guilty, then C is also guilty.
- C says: I am innocent and one of the others is guilty.

Answer each of the following questions about these testimonies:

1. Write down the propositional knowledge base describing the testimony of the three people, using the variables a, b, and c to represent whether or not a person is innocent (i.e., $a =$ `true` means a is innocent).
   - **A** $c \rightarrow \neg b$
   - **B** $\neg a \rightarrow \neg c$
   - **C** $c \wedge ((a \wedge \neg b) \vee (\neg a \wedge b))$
2. Write down a truth table for the knowledge base.

a	b	c	$c \rightarrow \neg b$	$\neg a \rightarrow \neg c$	$c \wedge ((a \wedge \neg b) \vee (\neg a \wedge b))$
T	T	T	F	T	F
T	T	F	T	T	F
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	T	F	T	F	F
F	F	T	T	T	F
F	F	F	T	T	F

3. Are the three testimonies consistent? Why or why not?

   Yes, these three testimonies are consistent. When a=True,B=False,c=True, they are True at the same time.

4. Assuming everyone is innocent (i.e., $a = b = c =$ `true`), who lied in their testimony?

   Given $a = True, b = True, c = True$. According to A, $c \rightarrow \neg b$ is false, so A lied. Also according to C, $c \wedge ((a \wedge \neg b) \vee (\neg a \wedge b)) = False$, C also lied.

5. Assuming all the testimony is true, who is innocent and who is guilty?

   A and C are innocent and B is guilty. According to the truth table above, the three testimonies are true only when a=True, b=False, and c=True

Problem 2

Programming Exercise: Implement a SAT Solver[1]

Part A

A *SAT-solver* is a program which decides whether or not a propositional sentence is *satisfiable*, i.e. has at least one solution. For example, if we have two logical variables a and b, then the sentence $a \lor b$ is satisfied by the assignment $a = \texttt{true}, b = \texttt{false}$.

A sentence is called a *clause* if it is a disjunction of literals, where a literal is a variable a or its negation $\neg a$. For example, $a \lor b$ is a clause. A sentence is in *conjunctive normal form* (CNF) if it conjunction of clauses, i.e. $(a \lor \neg b) \land (c \lor d)$ is in CNF. For this assignment, we would like you to implement a SAT-solver for CNF in a language of your choice. Here are some tips:

- Use integers to represent literals.
- Represent a clause as a list of integers, with positive integers representing non-negated literals and negative integers representing negated literals.

You can implement your solver however you want, so long as it is correct. You should submit your code along with a 1 paragraph description of the algorithm you wrote and why it works. Put your 1-paragraph solution in your PDF that you submit.

I used DPLL algorithm to build my SAT solver. The recurrence equation is

$$SAT(CNF) = SAT(CNF, i = True) \lor SAT(CNF, i = False)$$

This algorithm is based on brute force search to enumerate all possible combinations of boolean assignment for each variable. One optmization technique I used is that once a clause has a variable with value True, this clause is then removed from the rest of CNF to avoid extra computation since $True \land CNF$ will not affect the value of CNF. Also, early stopping is applied when a clause has only false values since $False \land CNF$ = $False$.

Part B

Check if the following sentences are SAT using your SAT-solver, and either (1) report the satisfying assignment, or (2) say that it is UNSAT:

- $(a \lor b \lor \neg c) \land (a \lor \neg d)$
- $(x \lor y \lor z) \land (x \lor y \lor \neg z) \land (x \lor \neg y \lor z) \land (x \lor \neg y \lor \neg z) \land (\neg x \lor y \lor z) \land (\neg x \lor y \lor \neg z) \land (\neg x \lor \neg y \lor z) \land (\neg x \lor \neg y \lor \neg z)$

1.

a	b	c	d
T	$*$	$*$	$*$
F	T	T	F
F	T	F	F
F	F	F	F

* represents any boolean assignment

2. UNSAT

[1] DPLL and other SAT-solving algorithms will be covered on Monday's lecture, but you are encouraged to think of your own solution before then.