CSCE 420 - Fall 2023
Homework 2 (HW2)
due: Tues, Oct 24, 11:59 pm

Turn-in answers as a Word document (HW2.docx or .pdf) and commit/push it to your class github repo.

When pushing the final version of your HW1, also run the command:
git tag "HW1" && git push origin "HW1"
This will be used to record time of submission for late penalty when applicable.

All homeworks must be typed, *not* hand-written and scanned/photo-shot.

1a. Prove that "Implication Introduction" (the opposite of Implication Elimination) is a **sound rule of inference** (ROI) using a **truth table**. If you have a Horn clause, with 1 positive literal and *n-1* negative literals, like (¬X∨Z∨¬Y), you can transform it into a conjunctive rule by collecting the negative literals as positive antecedents, e.g. X^Y→Z. It is sufficient to prove this for *n-1*=2 antecedents. (In fact, this is a truth-preserving operation, hence sound.)

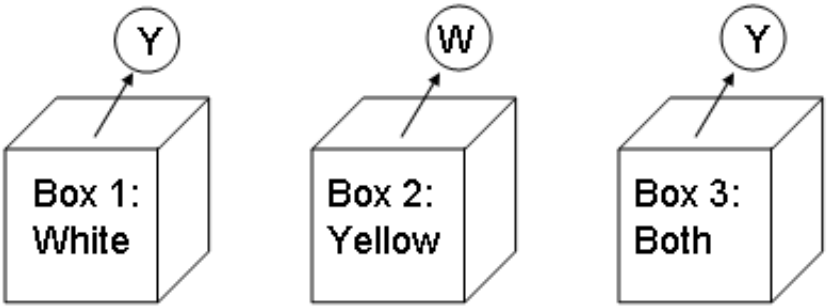
1b. Prove that (A^B→C^D) |- (A^B→C) ("conjunctive rule splitting") is a **sound rule-of-inference** using a **truth table**.

1c. Also prove (A^B→C^D) |= (A^B→C) using **Natural Deduction.** (hint: use 1a above)

1d. Also prove (A^B→C^D) |= (A^B→C) using **Resolution**.

2. **Sammy's Sport Shop**

You are the proprietor of *Sammy's Sport Shop*. You have just received a shipment of three boxes filled with tennis balls. One box contains only yellow tennis balls, one box contains only white tennis balls, and one contains both yellow and white tennis balls. You would like to stock the tennis balls in appropriate places on your shelves. Unfortunately, the boxes have been labeled incorrectly; the manufacturer tells you that you have exactly one box of each, but that **each box is definitely labeled wrong**. You draw one ball from each box and observe its color. Given the initial (incorrect) labeling of the boxes above, and the three observations, use Propositional Logic to infer the correct contents of the middle box.



Use propositional symbols in the following form: O1Y means a yellow ball was drawn (observed) from box 1, L1W means box 1 was initially labeled white, C1W means box 1 contains (only) white balls, and C1B means box 1 actually contains both types of tennis balls. Note, there is no 'O1B', etc, because you can't directly "observe both". When you draw a tennis ball, it will either be white or yellow.

The initial facts describing this particular situation are: {O1Y, L1W, O2W, L2Y, O3Y, L3B}

2a. Using these propositional symbols, write a propositional knowledge base (sammy.kb) that captures the knowledge in this domain (i.e. implications of what different observations or labels mean, as well as constraints inherent in this problem, such as that all boxes have different contents). *Do it in a complete and general way*, writing down *all* the rules and constraints, not just the ones needed to make the specific inference about the middle box. *Do not include derived knowledge* that depends on the particular labeling of this instance shown above; stick to what is stated in the problem description above. Your KB should be general enough to reason about any alternative scenario, not just the one given above (e.g. with different observations and labels and box contents).

2b. Prove that box 2 must contain white balls (**C2W**) using **Natural Deduction**.

2c. Convert your KB to CNF.

2d. Prove C2W using **Resolution.**

3. Do **Forward Chaining** for the *CanGetToWork* KB below.

You don't need to follow the formal FC algorithm (with agenda/queue and counts array). Just indicate which rules are triggered (in any order), and keep going until all consequences are generated.
Show the final list of all inferred propositions at the end. *Is CanGetToWork among them?*

KB = { a. CanBikeToWork → CanGetToWork
b. CanDriveToWork → CanGetToWork
c. CanWalkToWork → CanGetToWork
d. HaveBike ∧ WorkCloseToHome ^ Sunny → CanBikeToWork
e. HaveMountainBike → HaveBike
f. HaveTenSpeed → HaveBike
g. OwnCar → CanDriveToWork
h. OwnCar → MustGetAnnualInspection
i. OwnCar → MustHaveValidLicense
j. CanRentCar → CanDriveToWork
k. HaveMoney ∧ CarRentalOpen → CanRentCar
l. HertzOpen→ CarRentalOpen
m. AvisOpen→ CarRentalOpen
n. EnterpriseOpen→ CarRentalOpen
o. CarRentalOpen → IsNotAHoliday
p. HaveMoney ∧ TaxiAvailable → CanDriveToWork
q. Sunny ^ WorkCloseToHome → CanWalkToWork
r. HaveUmbrella ^ WorkCloseToHome → CanWalkToWork
s. Sunny → StreetsDry }

Facts: { Rainy, HaveMoutainBike, EnjoyPlayingSoccer, WorkForUniversity, WorkCloseToHome, HaveMoney, HertzClosed, AvisOpen, McDonaldsOpen }

4. Do **Backward Chaining** for the *CanGetToWork* KB.

In this case, you should follow the BC algorithm closely (the pseudocode for the propositional version of Back-chaining is given in the lecture slides).

Important: when you pop a subgoal (proposition) from the goal stack, you should systematically go through all rules that can be used to prove it IN THE ORDER THEY APPEAR IN THE KB. In some cases, this will lead to *back-tracking*, which you should show.

Also, the sequence of results depends on order in which antecedents are pushed onto the stack. If you have a rule like A^B→C, and you pop C off the stack, push the antecedents in reverse order, so B goes in first, then A; in the next iteration, A would be the next subgoal popped off the stack.