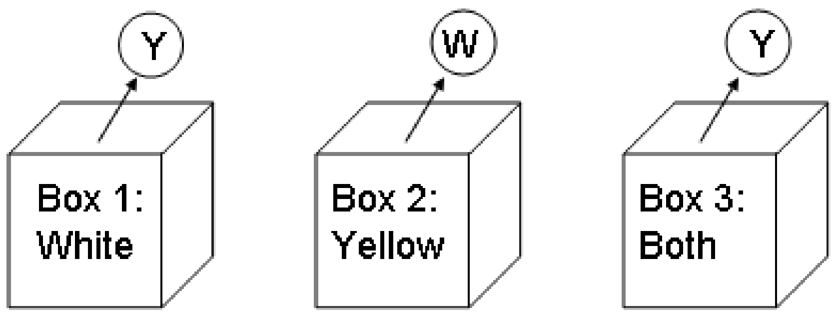
CSCE 420, Assignment A3

due: Tues, Mar 29, 3:55pm –
turn-in files by committing/pushing them to your TAMU github account (in directory A3/)

1. 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. One ball is drawn from each box and its color observed. Given the initial (incorrect) labeling of the boxes above, and the three observations, Propositional Logic can be used 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; You can't directly "observed both". When you draw a tennis ball, it will either be white or yellow. If white, the box could contain only white tennis balls, or balls of each color.

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

1a. Using these propositional symbols, write a script (in Python or C++) to generate a propositional knowledge base 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 each box is one of the colors, and all boxes have different colors; see the list a-e below). Do it in a complete and general way, writing down *all* the rules, not just the ones needed to make the specific inference about the middle box, and such that it could handle different observations or labels than the situation depicted above. To be systematic, write a script to generate the KB and number all the rules (similar to what was done for the wumpus world in the lecture slides). *Do not include derived knowledge*; focus on writing rules based only on information and constraints explicitly mentioned in the problem description above (see the list a-e below).

Your KB can be an ASCII file with propositional sentences formatted like this:

```
1. P v (Q ^ R)
2. A ^ -B -> C
3. -X v Y v Z
```

Here are sentences you need to translate into propositional logic to put in your KB, which can be used in the proof of C2W below (in 1b):

a. Every box contains one (and only) of the 3 colors.
b. Each box is a different color.
c. For each box and each color...if the box is labelled with that color, then it does not actually contain that color.
d. For each box and each color...if you observe a ball of that color drawn from that box, then it actually contains either that color alone, or both.
e. Include the 6 initial facts at the end.

1b. Prove that box 2 must contain white tennis balls (C2W) using Natural Deduction. (just write down each of the annotated steps in a text file)

1c. Re-write the KB in CNF (again, you will want to write a small script to generate all the clauses and number them; you don't have to show the conversion steps for each sentence).

1d. Prove C2W using Resolution Refutation. (write down each of the annotated steps in a text file)

Files to turn in (check into github):

- genKB_Sammy.py
- Sammy.KB.txt
- Sammy_C2W_proof.NatDed.txt
- gen_clauses_Sammy.py
- Sammy.CNF.txt
- Sammy_C2W_proof.Reso.txt

2. Consider the following set of conjunctive rules about ways to get to work:

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 }

2a. Show all the inferences that can be derived by Forward-Chaining. 'CanGetToWork' should be among them. Show the agenda at each step, show which rules are triggered, and indicate when new inferences are made (the first time they are inferred).

2b. Prove that 'CanGetToWork' is entailed by Backward-Chaining. Trace all steps, showing the goal stack, show which rules are used at each step, and indicate if and when back-tracking occurs.

Files to Turn in: CanGetToWork.backchain.txt, CanGetToWork.forward_chain.txt