CSCE 420 - Fall 2025

Homework 3 (HW3)

due: Tues, Oct 28, 11:59 pm

Turn-in answers as a PDF (HW3.pdf) and commit/push it to your class github repo.

All homeworks must be typed, ***not*** hand-written and scanned as photos.

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

A	B	C	D	A^B	C^D	A^B→C^D	A^B→C
T	T	T	T	T	T	T	T
T	T	T	F	T	F	F	T
T	T	F	T	T	F	F	F
T	T	F	F	T	F	F	F
T	F	T	T	F	T	T	T
T	F	T	F	F	F	T	T
T	F	F	T	F	F	T	T
T	F	F	F	F	F	T	T
F	T	T	T	F	T	T	T
F	T	T	F	F	F	T	T
F	T	F	T	F	F	T	T
F	T	F	F	F	F	T	T
F	F	T	T	F	T	T	T
F	F	T	F	F	F	T	T
F	F	F	T	F	F	T	T
F	F	F	F	F	F	T	T

Our table proves by modus ponens rule of inference that all models that satisfy the premises A^B→C^D, also satisfy the derived sentences A^B→C. I've highlighted all the matching truths in yellow.

1b. Also prove (A^B→C^D) |= (A^B→C) using **Natural Deduction.**

(Hint: it might help to use a ROI for "Implication Introduction". 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. This is a truth-preserving operation (hence sound), which you could prove to yourself using a truth table.)

1. Start with A^B→C^D
2. Modify into A^B→C using And Elimination (if C^D is true, then C must be true, so D can be removed)

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

Using the hint above, lets convert this sentence to CNF and resolve so we get a contradiction

Premises:

1. A^B→C^D turns into ¬A∨ (C^D) ∨¬B by Horn Clause implication elimination (used the hint)
2. ¬A∨ (C^D) ∨¬B turns into ¬A∨ C ∨¬B and ¬A∨ D ∨¬B by distribution

Conclusion negation:

1. A^B→C turns into ¬A∨C∨¬B via Horn Clause implication elimination (used the hint)
2. ¬A∨C∨¬B turns into A^¬C^ B via negation

Total Clauses:

1. (¬A∨ C ∨¬B)
2. (¬A∨ D ∨¬B)
3. A
4. ¬C
5. B

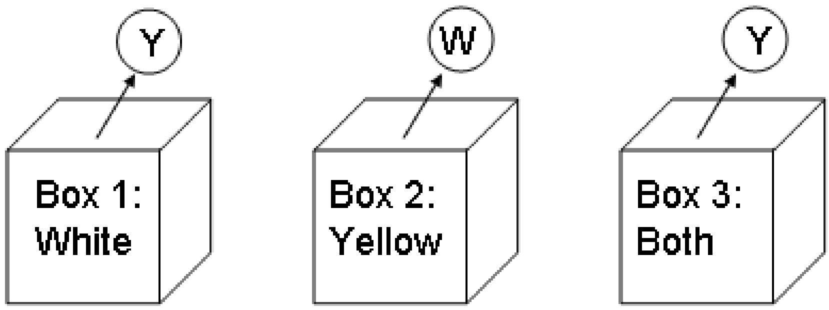
Steps:

6. Resolve 1 and 4 and turn it into ¬A∨ ¬B
7. Resolve 5 and 6 and turn it into ¬A
8. Resolve 3 and 7 and see that A and ¬A = ∅ empty set which is a CONTRADICTION
9. Resolve 3 and 6 and turn it into ¬B
10. Resolve 5 and 9 and see that B and ¬B= ∅ empty set which is a CONTRADICTION
11. Resolve 1 and 3 to get Cv¬B
12. Resolve 11 and 5 to get C
13. Resolve 12 and 4 to get C and ¬C= ∅ empty set which is a CONTRADICTION

We've proved entailment through resolution for all 3 conclusion variables :D

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).

Knowledge Base KB

Direct Observation:

1. O1Y
2. L1W
3. O2W
4. L2Y
5. O3Y
6. L3B

What we know from the labels being wrong:

7. (L1W → ¬C1W) since the label on box 1 is white, the color cannot be white because the label is wrong
8. (L2Y → ¬C2Y) same reasoning as above
9. (L3B → ¬C3B) same reasoning as above
10. (C1Y v C1B) box 1 has to be yellow or both
11. (C2W v C2B) same reasoning logic as above
12. (C3W v C3Y) same reasoning logic as above

Constraints that the boxes have on each other:

13. (C1Y → ¬C3Y) If box 1 is yellow then no other boxes can be yellow (in this case it only affects box 3 because box 2 is labeled wrong and isn't yellow balls)
14. (C1B → ¬C2B) If box 1 is both then no other boxes can be both

15. (C2W → ¬C3W) If box 2 is white then box 3 cannot be white

General Constraints (one color exists at a time):

16. (C1Y v C2Y v C3Y)
17. (C1W v C2W v C3W)
18. (C1B v C2B v C3B)
19. ¬ (C1Y^C3Y) Only one box c1 or c3 can be yellow at one time
20. ¬(C1B^C2B) same reasoning logic as above
21. ¬(C2W^C3W) same reasoning logic as above

What we know from the observed balls:

22. (O1Y→ (C1YvC1B) since the observed ball in box 1 is yellow then the resulting color must either be yellow or both
23. (O2W→ (C2WvC1B)
24. (O3Y→ (C1WvC1Y)
25. O1Y → ¬C1W
26. O2W → ¬C2Y
27. O3Y → ¬C3W

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

Premises: (I tried my best.)

1. O2W → ¬C2Y(rule 26)
2. O2W (rule 3)
3. Through modus ponens, we know that ¬C2Y is true
4. O3Y → ¬C3W(rule 27)
5. O3Y(rule 5)
6. Through modus ponens, we know that ¬C3W is true
7. L3B → ¬C3B (rule 9)
8. L3B (rule 6)
9. Through modus ponens, we know that ¬C3B is true
10. C3W v C3Y can be combined with premise 6 ¬C3W through disjunctive syllogism to conclude C3Y
11. Use rule 19 ¬ (C1Y^C3Y), since C3Y is true, we infer that C1Y must be false in order for the clause to be true
12. We can infer ¬C1Y through premise 11
13. (L1W → ¬C1W) rule 7
14. L1W rule 2
15. Through modus ponens, we know that ¬C1W is true
16. We can infer C1B is true through premise 12 premise 15
17. Using rule 20 ¬(C1B^C2B), this clause can only be true of C2B is false
18. We can infer ¬C2B through premise 17
19. (C2W v C2B) rule 11
20. Using disjunctive syllogism of premise 18 and 19, we can deduce that **C2W is true.**

2c. Convert your KB to CNF.

Direct Observation:

1. O1Y
2. L1W
3. O2W
4. L2Y
5. O3Y
6. L3B

What we know from the labels being wrong:

7. (L1W → ¬C1W) turns into (¬L1W v ¬C1W) via implication elimination
8. (L2Y → ¬C2Y) turns into (¬L2Y v ¬C2Y) via implication elimination
9. (L3B → ¬C3B) turns into (¬L3B v ¬C3B) via implication elimination

10. (C1Y v C1B)
11. (C2W v C2B)
12. (C3W v C3Y)

Constraints that the boxes have on each other:

13. (C1Y → ¬C3Y) turns into (¬C1Y v ¬C3Y) via implication elimination
14. (C1B → ¬C2B) turns into (¬C1B v ¬C2B) via implication elimination
15. (C2W → ¬C3W) turns into (¬C2W v ¬C3W) via implication elimination

General Constraints (one color exists at a time):

16. (C1Y v C2Y v C3Y)
17. (C1W v C2W v C3W)
18. (C1B v C2B v C3B)
19. ¬ (C1Y^C3Y) turns into (¬C1Y v ¬C3Y)via DeMorgan's Laws
20. ¬(C1B^C2B) turns into (¬C1B v ¬C2B)via DeMorgan's Laws
21. ¬(C2W^C3W) turns into (¬C2W v ¬C3W)via DeMorgan's Laws

What we know from the observed balls:

22. (O1Y→ (C1YvC1B) turns into ¬O1Y v C1Y v C1B via implication elimination
23. (O2W→ (C2WvC1B) turns into ¬O2W v C2W v C1B via implication elimination
24. (O3Y→ (C1WvC1Y) turns into ¬O3Y v C1W v C1Y via implication elimination
25. O1Y → ¬C1W turns into (¬O1Y v ¬C1W) via implication elimination
26. O2W → ¬C2Y turns into (¬O2W v ¬C2Y) via implication elimination
27. O3Y → ¬C3W turns into (¬O3Y v ¬C3W) via implication elimination

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

We can use all of the premises from above AND an additional clause/rule:

28. C2W conclusion needs to be negated into ¬C2W during resolution refutation

Steps:

29. Resolve rule 3 and rule 11 to get C2W v C1B
30. Resolve rule 11 and rule 16 to get C2B
31. Resolve rule 14 and rule 30 to get ¬C1B
32. Resolve rule 29 and rule 31 to get C2W
33. Resolve rule 32 C2W and rule 28 ¬C2W to get Ø empty set which is a Contradiction! Which proves C2W is true.

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, HaveMountainBike, EnjoyPlayingSoccer, WorkForUniversity, WorkCloseToHome, HaveMoney, HertzClosed, AvisOpen, McDonaldsOpen` }

We'll start with going through the facts to generate all consequences:

1. HaveMountainBike fires rule e HaveBike
2. AvisOpen fact fires rule m -> CarRentalOpen
3. CarRentalOpen fires rule o-> IsNotAHoliday
4. HaveMoney and CarRentalOpen fires rule k->CanRentCar
5. CanRentCar fires rule j ->CanDriveToWork
6. **6.** CanDriveToWork fires rule b-> CanGetToWork **(SOLUTION end)**

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.

Lets track backward chaining from CanGetToWork:
Steps:

1. Push **CanGetToWork** onto stack
2. Pop **CanGetToWork** and push CanBikeToWork (first rule in knowledge base)
3. Pop CanBikeToWork and push HaveBike, WorkCloseToHome, Sunny (antecedents are checked in reverse order)
4. Pop Sunny-> it fails->backtrack to CanGetToWork since none of the other conditions on stack need to be checked
5. Push CanDriveToWork
6. Pop CanDriveToWork and push OwnCar
7. Pop OwnCar->it fails-> backtrack to CanDriveToWork
8. Pop CanDriveToWork and push CanRentCar
9. Pop CanRentCar and push HaveMoney and CarRentalOpen (check antecedents in reverse order)
10. Pop CarRentalOpen and push HertzOpen
11. Pop HertzOpen-> it fails-> backtrack to CarRentalOpen
12. Pop CarRentalOpen and push AvisOpen
13. Pop AvisOpen-> it succeeds/is a fact-> move to HaveMoney
14. Pop HaveMoney-> it succeeds/is a fact
15. We then have proved all the variables for CanRentCar
16. We then have proved all the variables for CanDriveToWork
17. We then have proved all the variables for CanGetToWork