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 T positive literal and n-T negative literals, like (¬X v Z v ¬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-T=2 antecedents. (In fact, this is a truth-preserving operation, hence sound.)

X	Y	X^Y	Z	X^Y → Z
F	F	F	T	T
F	T	F	T	T
T	F	F	T	T
T	T	T	T	T

Because Z is always true, it does not matter what the truth value of X^Y is, the statement X^Y→Z is always true. Thus, if we have premises X and Y and derive (X ^ Y) → Z using Implication Introduction, Z is true on X and Y. Since the implication is always true, we can conclude that Implication Introduction is a sound ROI.

1b. 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
F	**F**	**F**	**F**	**F**	**F**	**T**	**T**
F	F	F	T	F	F	T	T
F	**F**	**T**	**F**	**F**	**F**	**T**	**T**
F	F	T	T	F	T	T	T
F	**T**	**F**	**F**	**F**	**F**	**T**	**T**
F	T	F	T	F	F	T	T
F	**T**	**T**	**F**	**F**	**F**	**T**	**T**
F	T	T	T	F	T	T	T
T	**F**	**F**	**F**	**F**	**F**	**T**	**T**
T	F	F	T	F	F	T	T
T	**F**	**T**	**F**	**F**	**F**	**T**	**T**
T	F	T	T	F	T	T	T
T	**T**	**F**	**F**	**T**	**F**	**F**	**F**
T	T	F	T	T	F	F	F
T	**T**	**T**	**F**	**T**	**F**	**F**	**T**
T	T	T	T	T	T	T	T

From this truth table, we can conclude that the statement **(A^B→C^D) |- (A^B→C)** is a sound ROI, as the conclusion is always true if the premise is true.

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

1. A^B (Assumption)
2. A^B → C^D (Premise)
3. C^D (Modus Ponens)
4. C (AndElimination)
5. A^B → C (Implication Introduction)

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

Convert to CNF

1. ¬(A^B) v C^D (Implication Elimination)
2. Distribute negation: (¬A v ¬B) v (C^D)
3. (¬A v ¬B v C)^(¬A v ¬B v D)
    a. (¬A v ¬B v C)
    b. (¬A v ¬B v D)

Negate the query

1. (IE) ¬(A ^ B) v C = ¬A v ¬B v C
2. ¬(¬A v ¬B v C) = A^B^¬C (DML)
    a. A
    b. B
    c. ¬C

Now we have:

1. A
2. B
3. ¬C
4. ¬A v ¬B v C
5. ¬A v ¬B v D
6. C (resolve 4 with 1 and 2)
7. ∅ (Resolve 6 and 3, empty clause)

2a.

Initial facts: O1Y, L1W, O2W, L2Y, O3Y, L3B

Using x to represent a color in our domain (W, Y, B).

Using z to represent a box in our domain (1, 2, 3)

KB = {

Boxes are definitely labeled wrong

1. L1Y → ¬C1Y
2. L1W → ¬C1W
3. L1B → ¬C1B
4. L2Y → ¬C2Y
5. L2W → ¬C2W
6. L2B → ¬C2B
7. L3Y → ¬C3Y
8. L3W → ¬C3W
9. L3B → ¬C3B

If we observe a color out of a particular box, the box is that color or both

10. O1Y → C1Y v C1B
11. O1W → C1W v C1B
12. O2Y → C2Y v C2B
13. O2W → C2W v C2B
14. O3Y → C3Y v C3B
15. O3W → C3W v C3B

Colors not the same between boxes

16. C1Y → ¬C2Y^¬C3Y
17. C1W → ¬C2W^¬C3W
18. C1B → ¬C2B^¬C3B
19. C2Y → ¬C1Y^¬C3Y
20. C2W → ¬C1W^¬C3W
21. C2B → ¬C1B^¬C3B
22. C3Y → ¬C1Y^¬C2Y
23. C3W → ¬C1W^¬C2W
24. C3B → ¬C1B^¬C2B

If a box is not two distinct colors, it must be the remaining color

25. ¬C1W^¬C1Y → C1B
26. ¬C1W^¬C1B → C1Y
27. ¬C1Y^¬C1B → C1W
28. ¬C2W^¬C2Y → C2B
29. ¬C2W^¬C2B → C2Y
30. ¬C2Y^¬C2B → C2W
31. ¬C3W^¬C3Y → C3B
32. ¬C3W^¬C3B → C3Y
33. ¬C3Y^¬C3B → C3W

If we know the colors (x_1, x_2) of two boxes (z_1 and z_2), then the last box must be the remaining color (z_3 is x_3)

34. Cz_1x_1^ $Cz_2x_2 \rightarrow Cz_3x_3$
    a. i.e. C1Y^C2W → C3B

}

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

Premises	**Derivations**
35. O1Y	(35, 10, and Modus Ponens) 41. C1Y v C1B
36. L1W	(36, 2, and MP) 42. ¬C1W
37. O2W	(37, 13, MP) 43. C2W v C2B
38. L2Y	(38, 4, MP) 44. ¬C2Y
39. O3Y	(39, 14, MP) 45. C3Y v C3B
40. L3B	(40, 9, MP) 46. ¬C3B

	(40, 39, AndIntroduction) 47. ¬C3B^(C3Y v C3B) = ¬C3B^C3Y 48. (AndElimination) C3Y
	(22, MP) 49. ¬C1Y^¬C2Y 50. (49, AE) ¬C1Y 51. (50, 42, AI) ¬C1Y^¬C1W 52. (51, 25, MP) C1B 53. (48, 52, AI) C1B^C3Y 54. (34, 53, MP) C2W

2c. Convert your KB to CNF.

Converted KB using Implication Elimination for every rule (each number corresponds to the original KB rule)

KB$_{CNF}$ = {

1. ¬L1Y v ¬C1Y
2. ¬L1W v ¬C1W
3. ¬L1B v ¬C1B
4. ¬L2Y v ¬C2Y
5. ¬L2W v ¬C2W
6. ¬L2B v ¬C2B
7. ¬L3Y v ¬C3Y
8. ¬L3W v ¬C3W
9. ¬L3B v ¬C3B
10. ¬O1Y v C1Y v C1B
11. ¬O1W v C1W v C1B
12. ¬O2Y v C2Y v C2B
13. ¬O2W v C2W v C2B
14. ¬O3Y v C3Y v C3B
15. ¬O3W v C3W v C3B
16. ¬C1Y v (¬C2Y^¬C3Y) = (¬C1Y v ¬C2Y)^(¬C1Y v ¬C3Y)
    a. (¬C1Y v ¬C2Y)
    b. (¬C1Y v ¬C3Y)
17. (¬C1W v ¬C2W)^(¬C1W v ¬C3W)
    a. (¬C1W v ¬C2W)
    b. (¬C1W v ¬C3W)
18. (¬C1B v ¬C2B)^(¬C1B v ¬C3B)
    a. (¬C1B v ¬C2B)
    b. (¬C1B v ¬C3B)
19. ¬C2Y v (¬C1Y^¬C3Y) = (¬C2Y v ¬C1Y)^(¬C2Y v ¬C3Y)
    a. (¬C2Y v ¬C1Y)
    b. (¬C2Y v ¬C3Y)
20. (¬C2W v ¬C1W)^(¬C2W v ¬C3W)

    a. (¬C2W v ¬C1W)
    b. (¬C2W v ¬C3W)
21. (¬C2B v ¬C1B)^(¬C2B v ¬C3B)
    a. (¬C2B v ¬C1B)
    b. (¬C2B v ¬C3B)
22. (¬C3Y v ¬C1Y)^(¬C3Y v ¬C2Y)
    a. (¬C3Y v ¬C1Y)
    b. (¬C3Y v ¬C2Y)
23. (¬C3W v ¬C1W)^(¬C3W v ¬C2W)
    a. (¬C3W v ¬C1W)
    b. (¬C3W v ¬C2W)
24. (¬C3B v ¬C1B)^(¬C3B v ¬C2B)
    a. (¬C3B v ¬C1B)
    b. (¬C3B v ¬C2B)
25. ¬(¬C1W^¬C1Y) v C1B = (DML) C1W v C1Y v C1B
26. C1W v C1Y v C1B
27. C1W v C1Y v C1B
28. C2W v C2Y v C2B
29. C2W v C2Y v C2B
30. C2W v C2Y v C2B
31. C3W v C3Y v C3B
32. C3W v C3Y v C3B
33. C3W v C3Y v C3B
34. $\neg Cz_1x_1 \vee \neg Cz_2x_2 \vee Cz_3x_3$
35. (Negate query) ¬C2W

}

2d. Prove C2W using Resolution.

Premise	Derivation
36. O1Y	(36, 10, AI) 43. C1Y v C1B
37. L1W	(37, 2, AI) 44. ¬C1W
38. O2W	(38, 13, AI) 45. C2W v C2B
39. L2Y	(39, 4, AI) 46. ¬C2Y
40. O3Y	(40, 14, AI) 47. C3Y v C3B
41. L3B	(41, 9, AI) 48. ¬C3B
42. ¬C2W	(48, 47, AI) 49. C3Y
	(22a, 49, AI) 50. ¬C1Y
	(50, 43, AI) 51. C1B
	(18a, 51, AI) 52. ¬C2B
	(52, 45, AI) 53. C2W
	(53, 42, AI) 54. ∅

3. Do Forward Chaining for the CanGetToWork KB below

Facts (with irrelevant facts discarded): { HaveMoutainBike, WorkCloseToHome, HaveMoney, AvisOpen **}**

Start	HaveMoutainBike, WorkCloseToHome, HaveMoney, AvisOpen
Rule e	HaveMoutainBike, WorkCloseToHome, HaveMoney, AvisOpen, **HaveBike**
Rule m	WorkCloseToHome, HaveMoney, AvisOpen, HaveBike, **CarRentalOpen**
Rule k, Rule o	WorkCloseToHome, HaveMoney, AvisOpen, HaveBike, CarRentalOpen, **CanRentCar, IsNotAHoliday**
Rule j	WorkCloseToHome, HaveMoney, AvisOpen, HaveBike, CarRentalOpen, CanRentCar, IsNotAHoliday, **CanDriveToWork,**
Rule b	WorkCloseToHome, HaveMoney, AvisOpen, HaveBike, CarRentalOpen, CanRentCar, IsNotAHoliday, CanDriveToWork, **CanGetToWork**

CanGetToWork is in the final list of inferred propositions

4. Do Backward Chaining for the CanGetToWork KB

Facts (with irrelevant facts discarded): { HaveMountainBike, WorkCloseToHome, HaveMoney, AvisOpen **}**

Start	**CanGetToWork**
Pop CanGetToWork, replace with antecedent of rule (a)	CanBikeToWork
Pop CanBikeToWork, replace with antecedent of rule (d)	HaveBike, WorkCloseToHome, Sunny
Pop HaveBike, push HaveMountainBike (rule e)	HaveMountainBike, WorkCloseToHome, Sunny
Pop HaveMountainBike (fact)	WorkCloseToHome, Sunny
Pop WorkCloseToHome (fact)	Sunny
Pop Sunny, push StreetsDry (rule s)	StreetsDry
StreetsDry not provable, backtrack to other rule for CanGetToWork (rule b)	CanDriveToWork
Pop CanDriveToWork, push OwnCar (g)	OwnCar
OwnCar not provable, backtrack to other rule for CanDriveToWork (rule j)	
Push CanRentCar (j)	CanRentCar
Pop CanRentCar, push HaveMoney, CarRentalOpen (rule k)	HaveMoney, CarRentalOpen
Pop HaveMoney (fact)	CarRentalOpen
Pop CarRentalOpen, push HertzOpen	HertzOpen
HertzOpen not provable, backtrack to other rule for CarRentalOpen (rule m)	
Push AvisOpen (m)	AvisOpen
Pop AvisOpen (fact)	**Return True**