Octavio Almanza
UIN: 628005492

Homework 2

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 $(\neg X \vee Z \vee \neg Y)$, you can transform it into a conjunctive rule by collecting the negative literals as positive antecedents, e.g. $X \wedge Y \Rightarrow Z$. It is sufficient to prove this for $n-1=2$ antecedents. (In fact, this is a truth-preserving operation, hence sound.)

Recall: A sound rule of inference only generates new sentences that are entailed.

X	Y	Z	¬X	¬Y	¬X∨¬Y∨Z	X∧Y	X∧Y⇒Z
F	F	F	T	T	**T**	F	**T**
F	F	T	T	T	**T**	F	**T**
F	T	F	T	F	**T**	F	**T**
F	T	T	T	F	**T**	F	**T**
T	F	F	F	T	**T**	F	**T**
T	F	T	F	T	**T**	F	**T**
T	T	F	F	F	F	T	F
T	T	T	F	F	**T**	T	**T**

Table 1: Truth Table for 1a

Since all models that satisfy the premise ¬X∨¬Y∨Z also satisfy the conclusion X∧Y⇒Z, we prove that the conclusion is entailed. Thus, implication introduction is a sound rule of inference. In fact, since the premise and the conclusion are satisfied by the exact same models, implication introduction is a truth-preserving rule of inference.

1b. Prove that $(A \wedge B \Rightarrow C \wedge D) \vdash (A \wedge B \Rightarrow C)$ ("conjunctive rule splitting") is a sound rule-of-inference using a truth table.

Recall: A sound rule of inference only generates new sentences that are entailed.

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**

Table 2: Truth Table for 1b

Since all models that satisfy the premise A∧B⇒C∧D also satisfy the conclusion A∧B⇒C, we prove that the conclusion is entailed. Thus, conjunctive rule splitting is a sound rule-of-inference.

1c. Also prove $(A \wedge B \Rightarrow C \wedge D) \models (A \wedge B \Rightarrow C)$ using Natural Deduction. (hint: use 1a above)

Premise

1. A∧B⇒C∧D

Derivations

2. ¬(A∧B)∨C∧D [Implication Elimination, 1]
3. (¬A∨¬B)∨C∧D [DeMorgan's Law, 2]
4. (¬A∨¬B∨C)∧(¬A∨¬B∨D) [Distributivity, 3]
5. (¬A∨¬B∨C) [And Elimination, 4]
6. A∧B⇒C [Implication Introduction, 5]

1d. Also prove $(A \wedge B \Rightarrow C \wedge D) \models (A \wedge B \Rightarrow C)$ using Resolution.

Convert premise to Conjunctive-Normal Form Clauses

1. A∧B⇒C∧D
2. ¬(A∧B)∨C∧D [Implication Elimination, 1]
3. (¬A∨¬B)∨C∧D [DeMorgan's Law, 2]
4. (¬A∨¬B∨C)∧(¬A∨¬B∨D) [Distributivity, 3]
5. ¬A∨¬B∨C [And Elimination, 4] (CNF)
6. ¬A∨¬B∨D [And Elimination, 4] (CNF)

Negate The Query:

- ¬(A∧B⇒C)
- ¬(¬(A∧B)∨C) [Implication Elimination]
- ¬¬(A∧B)∧¬C [DeMorgan's Laws]
- A∧B∧¬C [Double Negation Elimination]
- A, B, ¬C [And Elimination]

Apply resolution refutation by adding the negated query into the premises:

Premises (All in CNF):

1. ¬A∨¬B∨C
2. ¬A∨¬B∨D
3. A
4. B
5. ¬C

Derivations

6. ¬B∨C [Resolution on 1 and 3]
7. C [Resolution on 4 and 6]
8. ∅ [Resolution on 5 and 7]

2. Sammy's Sport Shop

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).

Label Implications

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

Color Implications

10. $L1W \Leftrightarrow \neg L2W \wedge \neg L3W$
11. $L1Y \Leftrightarrow \neg L2Y \wedge \neg L3Y$
12. $L1B \Leftrightarrow \neg L2B \wedge \neg L3B$
13. $L2W \Leftrightarrow \neg L1W \wedge \neg L3W$
14. $L2Y \Leftrightarrow \neg L1Y \wedge \neg L3Y$
15. $L2B \Leftrightarrow \neg L1B \wedge \neg L3B$
16. $L3W \Leftrightarrow \neg L2W \wedge \neg L1W$
17. $L3Y \Leftrightarrow \neg L2Y \wedge \neg L1Y$
18. $L3B \Leftrightarrow \neg L2B \wedge \neg L1B$
19. $\neg C1Y \wedge \neg C1B \Leftrightarrow C1W$
20. $\neg C1W \wedge \neg C1B \Leftrightarrow C1Y$
21. $\neg C1Y \wedge \neg C1W \Leftrightarrow C1B$
22. $\neg C2Y \wedge \neg C2B \Leftrightarrow C2W$
23. $\neg C2W \wedge \neg C2B \Leftrightarrow C2Y$
24. $\neg C2Y \wedge \neg C2W \Leftrightarrow C2B$
25. $\neg C3Y \wedge \neg C3B \Leftrightarrow C3W$
26. $\neg C3W \wedge \neg C3B \Leftrightarrow C3Y$
27. $\neg C3Y \wedge \neg C3W \Leftrightarrow C3B$

Observation Implications

28. $O1W \Rightarrow C1W \vee C1B$
29. $O1Y \Rightarrow C1Y \vee C1B$
30. $O2W \Rightarrow C2W \vee C2B$
31. $O2Y \Rightarrow C2Y \vee C2B$
32. $O3W \Rightarrow C3W \vee C3B$
33. $O3Y \Rightarrow C3Y \vee C3B$
34. $O1W \Rightarrow \neg C1Y$
35. $O1Y \Rightarrow \neg C1W$
36. $O2W \Rightarrow \neg C2Y$
37. $O2Y \Rightarrow \neg C2W$
38. $O3W \Rightarrow \neg C3Y$
39. $O3Y \Rightarrow \neg C3W$

Contains Implications

40. $C1W \Leftrightarrow \neg C2W \wedge \neg C3W$
41. $C1Y \Leftrightarrow \neg C2Y \wedge \neg C3Y$
42. $C1B \Leftrightarrow \neg C2B \wedge \neg C3B$
43. $C2W \Leftrightarrow \neg C1W \wedge \neg C3W$
44. $C2Y \Leftrightarrow \neg C1Y \wedge \neg C3Y$
45. $C2B \Leftrightarrow \neg C1B \wedge \neg C3B$
46. $C3W \Leftrightarrow \neg C2W \wedge \neg C1W$
47. $C3Y \Leftrightarrow \neg C2Y \wedge \neg C1Y$
48. $C3B \Leftrightarrow \neg C2B \wedge \neg C1B$
49. $C1W \Rightarrow O1W$
50. $C1Y \Rightarrow O1Y$
51. $C1B \Rightarrow O1W \vee O1Y$
52. $C2W \Rightarrow O2W$
53. $C2Y \Rightarrow O2Y$
54. $C2B \Rightarrow O2W \vee O2Y$
55. $C3W \Rightarrow O3W$
56. $C3Y \Rightarrow O3Y$
57. $C3B \Rightarrow O3W \vee O3Y$

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

Initial Facts

58. L1W
59. L2Y
60. L3B
61. O1Y
62. O2W
63. O3Y

Derivations

64. ¬C3B [Modus Ponens with rules 60 and 9]
65. C3Y∨C3B [Modus Ponens with rules 33 and 63]
66. C3Y [Resolution with rules 64 and 65]
67. ¬C2Y∧¬C1Y [Modus Ponens with rules 66 and 47]
68. ¬C2Y[And Elimination on rule 67]
69. ¬C1Y[And Elimination on rule 67]
70. ¬C1W [Modus Ponens with rules 1 and 58]
71. ¬C1Y∧¬C1W [And introduction with rules 69 and 70]
72. C1B [Modus Ponens with rules 71 21]
73. ¬C2B∧¬C3B [Modus Ponens with rules 72 and 42]
74. ¬C2B [And Elimination on rule 73]
75. ¬C2Y∧¬C2B [And Introduction on rules 68 and 74]
76. **C2W [Modus Ponens on rules 75 and 22]**

2c. Convert your KB to CNF

Label Implications

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

Color Implications

10. L1W∨L2W∨L3W
11. L1Y∨L2Y∨L3Y
12. L1B∨L2B∨L3B
13. ¬L1W∨¬L2W
14. ¬L1W∨¬L3W
15. ¬L1Y∨¬L2Y
16. ¬L1Y∨¬L3Y
17. ¬L1B∨¬L2B
18. ¬L1B∨¬L3B
19. ¬L2W∨¬L3W
20. ¬L2Y∨¬L3Y
21. ¬L2B∨¬L3B
22. C1W∨C1Y∨C1B
23. C2W∨C2Y∨C2B
24. C3W∨C3Y∨C3B
25. ¬C1W∨¬C1Y
26. ¬C1W∨¬C1B
27. ¬C1Y∨¬C1B
28. ¬C2W∨¬C2Y
29. ¬C2W∨¬C2B
30. ¬C2Y∨¬C2B
31. ¬C3W∨¬C3Y
32. ¬C3W∨¬C3B
33. ¬C3Y∨¬C3B

Observation Implications

34. ¬O1W∨C1W∨C1B
35. ¬O1Y∨C1Y∨C1B
36. ¬O2W∨C2W∨C2B
37. ¬O2Y∨C2Y∨C2B
38. ¬O3W∨C3W∨C3B
39. ¬O3Y∨C3Y∨C3B
40. ¬O1W∨¬C1Y
41. ¬O1Y∨¬C1W
42. ¬O2W∨¬C2Y
43. ¬O2Y∨¬C2W
44. ¬O3W∨¬C3Y
45. ¬O3Y∨¬C3W

Contains Implications

46. C1W∨C2W∨C3W
47. C1Y∨C2Y∨C3Y
48. C1B∨C2B∨C3B
49. ¬C1W∨¬C2W
50. ¬C1W∨¬C3W
51. ¬C1Y∨¬C2Y
52. ¬C1Y∨¬C3Y
53. ¬C1B∨¬C2B
54. ¬C1B∨¬C3B
55. ¬C2W∨¬C3W
56. ¬C2Y∨¬C3Y
57. ¬C2B∨¬C3B
58. ¬C1W∨O1W
59. ¬C1Y∨O1Y
60. ¬C1B∨O1Y∨O1W
61. ¬C2W∨O2W
62. ¬C2Y∨O2Y
63. ¬C2B∨O2Y∨O2W
64. ¬C3W∨O3W
65. ¬C3Y∨O3Y
66. ¬C3B∨O3Y∨O3W

2d. Prove C2W using **Resolution**

Premises

67. L1W
68. L2Y
69. L3B
70. O1Y
71. O2W
72. O3Y

Derivations

73. ¬C2W [Negated query]
74. C2Y∨C2B [Resolution between 73 and 23]
75. ¬L2Y∨C2B [Resolution with 74 and 5]
76. C2B [Resolution with 75 and 68]
77. ¬C1B [Resolution with 76 and 53]
78. C1W∨C1Y [Resolution with 77 and 22]
79. ¬L1W∨C1Y [Resolution with 78 and 1]
80. C1Y [Resolution with 79 and 67]
81. ¬C3Y [Resolution with 80 and 52]
82. ¬O3Y∨C3B [Resolution with 81 and 39]
83. C3B [Resolution with 82 and 72]
84. ¬L3B [Resolution with 83 and 9]
85. ∅ [Resolution with 84 and 69]

Since we were able to derive the empty clause ∅ from the premise ¬C2W, we have demonstrated that the clause ¬C2W causes a contradiction in Sammy.kb, which consequently proves C2W.

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}

Note: For the purpose of this problem, I will discard all of the literals in **Facts** that do not appear in the knowledge base. I was given permission to do this in Campuswire Post #55.

1. Initialize **Queue** and **Inferred** data structures.
   - **Queue** = {HaveMountainBike, WorkCloseToHome, HaveMoney, AvisOpen}
   - **Inferred** = {}
2. Pop HaveMountainBike. HaveMountainBike triggers **rule e**. Add HaveMountainBike to **Inferred** and HaveBike to the **Queue**.
   - **Queue** = {WorkCloseToHome, HaveMoney, AvisOpen, HaveBike}
   - **Inferred** = {HaveMountainBike}
3. Pop WorkCloseToHome. WorkCloseToHome and our current inference list does not trigger any other rules.
   - **Queue** = {HaveMoney, AvisOpen, HaveBike}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome}

4. Pop HaveMoney. HaveMoney and our current inference list does not trigger any other rules.
   - **Queue** = {AvisOpen, HaveBike}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome, HaveMoney}
5. Pop AvisOpen. AvisOpen and HaveMoney trigger **rule k**. Add AvisOpen to **Inferred** and CanRentCar to the **Queue**.
   - **Queue** = {HaveBike, CanRentCar}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome, HaveMoney, AvisOpen}
6. Pop HaveBike. HaveBike and our current inference list does not trigger any other rules.
   - **Queue** = {CanRentCar}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome, HaveMoney, HaveBike}
7. Pop CanRentCar. CanRentCar triggers **rule j**. Add CanRentCar to **Inferred** and CanDriveToWork to the **Queue**.
   - **Queue** = {CanDriveToWork}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome, HaveMoney, HaveBike, CanRentCar}
8. Pop CanDriveToWork. CanDriveToWork triggers **rule b**. Add CanDriveToWork to **Inferred** and CanGetToWork to the **Queue**.
   - **Queue** = {CanGetToWork}
   - **Inferred** = {HaveMountainBike, WorkCloseToHome, HaveMoney, HaveBike, CanRentCar, CanDriveToWork}
9. Pop CanGetToWork. **CanGetToWork is our query, so we are done!**

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.

Note: In order for the tracing to terminate with the conclusion, we must push the following known facts into the KB:

{t. HaveMountainBike, u. WorkCloseToHome, v. HaveMoney, w. AvisOpen}

1. Initialize the goal stack and push the query CanGetToWork.
   - Goal Stack = {CanGetToWork}
2. Pop CanGetToWork. We branch by pushing the antecedents in rule a (Could have used a, b, or c)
   - Goal Stack = {CanBikeToWork}
3. Pop CanBikeToWork. We push the antecedents in rule d
   - Goal Stack = {HaveBike, WorkCloseToHome, Sunny}
4. Pop HaveBike. Push the antecedents in rule e (could have used e or f)
   - Goal Stack = {WorkCloseToHome, Sunny, HaveMountainBike}
5. Pop WorkCloseToHome. It is a known fact, so we don't push any antecedents.
   - Goal Stack = {Sunny, HaveMountainBike}
6. Pop Sunny. Sunny not provable, so we backtrack our choice in step 4 and push the antecedents of rule f instead.
   - Goal Stack = {WorkCloseToHome, Sunny, HaveTenSpeed}
7. Pop WorkCloseToHome. It is a known fact (rule u), so we don't push any antecedents.
   - Goal Stack = {Sunny, HaveTenSpeed}
8. Pop Sunny. Sunny is not provable and since we have no other antecedents to push for HaveBike, we backtrack to **Step 2** and pick a different rule for the antecedents of CanGetToWork. We select rule b and push CanDriveToWork to the Goal Stack.
   - Goal Stack = {CanDriveToWork}

9. Pop CanDriveToWork. We branch by pushing the antecedents in rule g (could have used g, j, p, or q)
   - Goal Stack = {OwnCar}
10. Pop OwnCar. OwnCar is not provable, so we backtrack to **Step 9** and push the antecedents in rule j instead.
    - Goal Stack = {CanRentCar}
11. Pop CanRentCar. Push the antecedents in rule k.
    - Goal Stack = {HaveMoney, CarRentalOpen}
12. Pop HaveMoney. HaveMoney is a known fact (rule v), so we continue.
    - Goal Stack = {CarRentalOpen}
13. Pop CarRentalOpen. We branch by pushing the antecedents in rule l (could have used l, m, or n)
    - Goal Stack = {HertzOpen}
14. Pop HertzOpen. HertzOpen is not provable, so we backtrack to **Step 13** and push the antecedents in rule m instead.
    - Goal Stack = {AvisOpen}
15. Pop AvisOpen. AvisOpen is a known fact, so we continue.
    - Goal Stack = {}
16. **Since the Goal Stack is empty, we have demonstrated that we "CanGetToWork" with our knowledge base and facts.**