HOMEWORK 2

Question 1: Prove $(A \wedge B \rightarrow C \wedge D) \vdash (A \wedge B \rightarrow C)$ using truth table.

A	B	C	D	C^D	A^B	A^B->C^D	A^B->C
T	T	T	T	T	T	T	T
T	T	T	F	F	T	F	T
T	T	F	T	F	T	F	F
T	T	F	F	F	T	F	F
T	F	T	T	T	F	T	T
T	F	T	F	F	F	T	T
T	F	F	T	F	F	T	F
T	F	F	F	F	F	T	F
F	T	T	T	T	F	T	T
F	T	T	F	F	F	T	T
F	T	F	T	F	F	T	F
F	T	F	F	F	F	T	F
F	F	T	T	T	F	T	T
F	F	T	F	F	F	T	T
F	F	F	T	F	F	T	F
F	F	F	F	F	F	T	F

1b. Prove $(A \wedge B \rightarrow C \wedge D) \models (A \wedge B \rightarrow C)$ using NATURAL DEDUCTION.

Premises

1. $A \wedge B$
2. $C \wedge D$
3. $A \wedge B \rightarrow C \wedge D$

Derive

4. A (AE, 1)
5. C (AE, 2)
6. B (MP, 1, 5)
7. D (MP,
8. $A \wedge C$
9. $B \wedge C$
10. $A \wedge D$
11. $(\neg A \vee \neg B) \vee (C \wedge D)$ (IE, 3)
12. $(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee D)$ (Distributivity)
13. $(\neg A \vee \neg B \vee C)$ (AE)
14. $A \wedge B \rightarrow C$ (implication intro, 13)

1c. Prove $(A \wedge B \rightarrow C \wedge D) \models (A \wedge B \rightarrow C)$ using Resolution.

CNF: $(A \wedge B) \rightarrow (C \wedge D)$

1. $\neg(A \wedge B) \vee (C \wedge D) == (\neg A \vee \neg B) \vee (C \wedge D) == (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee D)$
2. $(\neg A \vee \neg B \vee C)$ $\quad \neg(\neg A \vee \neg B \vee C)$
3. $(\neg A \vee \neg B \vee D)$
4. $(A \vee B \vee \neg C)$
5. $(B \vee D)$
6. $\neg B$
7. D
8. $\neg C$
9. $()$

$A \wedge B \wedge C \rightarrow A \wedge B \rightarrow C$ (IE)

$\rightarrow \neg A \vee \neg B \vee C$

$\rightarrow (A \wedge B \rightarrow C \wedge D) \models (A \wedge B) \rightarrow C$

Question 2:

- $O1Y \to C1Y \vee C1B$ (a)
- $O2W \to C2W \vee C2B$ (b)
- $O3Y \to C3Y \vee C3B$ (c)

- Since all labels are wrong:
  - $L1W \to \neg C1W$, $L2Y \to \neg C2Y$, $L3B \to \neg C3B$. (d)
- There is at least 1 box of each color: $C1Y \vee C1B$, $C2W \vee C2B$, $C3Y \vee C3W$ (e)
- no 2 boxes have same content.

2b. Prove C2W using natural deduction.

1. $C3Y \vee C3B$ (mp, O3Y)
2. $C3Y$ (from (1) and (e))
3. $\neg C1Y \wedge \neg C2Y$ (mp, (3), no same color)
4. $C1Y \vee C1B$ (mp, O1Y, (a))
5. $\neg C1Y$ (3, And Elim)
6. $C1B$ (Resolution, 4, 5)
7. $C2W \vee C2B$ (mp, O2W, (b))
8. $\neg C2B \wedge \neg C3B$ (6, no same color)
9. $\neg C2B$ (8)
10. $C2W$ (7, 9, resolution).

2b. Convert your KB to CNF.

- $O3Y \to C3Y \vee C3B$ → 0. $\neg O3Y \vee C3Y \vee C3B$
- $O1Y \to C1Y \vee C1B$ → 1. $\neg O1Y \vee C1Y \vee C1B$.
- $O2W \to C2W \vee C2B$ → 2. $\neg O2W \vee C2W \vee C2B$.
- $L3B \to \neg C3B$ → 3. $\neg L3B \vee \neg C3B$.
- $C1B \to \neg C2B \wedge \neg C3B$ → 4a. $\neg C1B \vee \neg C2B$, 4b. $\neg C1B \vee \neg C3B$.
- $C3Y \to \neg C2Y \wedge \neg C1Y$ → 5a. $\neg C3Y \vee \neg C2Y$, 5b. $\neg C3Y \vee \neg C1Y$.

6. O1Y
7. O2W
8. O3Y
9. L1W
10. L2Y
11. L3B
12. $\neg C2W$ (negation)
13. $C3Y \vee C3B$ (res, 8, 10)
14. $\neg C3B$ (res, 11, 3)
15. $C3Y$ (res, 13, 14)
16. $\neg C1Y$ (res, 15, 5b)
17. $C1Y \vee C1B$ (res, 6, 1)
18. $C1B$ (res, 16, 17)
19. $\neg C2B$ (res, 18, 4a)
20. $C2W \vee C2B$ (res, 7, 2)
21. $C2W$ (res, 19, 20)
22. $\square$ (res, 21, 12)

. Question 3: Do Forward chaining.

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

Prove: CANGETTOWORK

. KB Facts:

* Proof by Forward Chaining.

a. GetToWork(bike) → AtWork(yes)

b. GetToWork(drive) → AtWork(yes)

c. GetToWork(walk) → AtWork(yes)

d. Bike(own) ∧ distance(close) ∧ Weather(sunny) → GetToWork(bike)

e. Own(mountainbike) → Own(bike)

f. Own(TenSpeed) → Own(bike)

g. Own(Car) → GetToWork(Drive)

h. Own(car) → MustGet(Annual inspection)

i. Own(car) → musthave(Valid license)

j. Rent(Car) → GetToWalk(Drive)

k. Own(money) ∧ Open(Car rental) → Rent(car)

l. Open(Hertz) → Open(Car Rental)

m. AvisOpen → Open(car rental).

n. Open(Enterprise) → Open(car rental)

o. Open(Car rental) → holiday(no)

p. Own(money) ∧ open(taxi) → gettowork(drive)

q. Weather(sunny) ∧ Distance(close) → GetToWork(Walk)

. Given facts:

. Weather(rainy) → ~~d~~ → ~~q~~ → ~~s~~

. Own(mountainbike) → Own(bike)

. Enjoy(play soccer)

. Work(uni)

. Distance(close)

. Own(money)

. Closed(hertz) → ~~l~~

. Open(Avis) → Open(CarRental) ∧ Own(money) → Rent(Car) → GetToWork(Drive)

. Open(McDonalds) → AtWork(Yes)

we can get the option 'CANGETTOWORK' from this statement.

Question 4: Do Backward Chaining.

We start with the goal CanGetToWork and push it onto the goal stack.
We pop CanGetToWork from the goal stack and look for rules that can be used to prove it.
Rules a,b, and c can be used to prove CanGetToWork.
We first try rule a, which has the antecedent CanBikeToWork. We push CanBikeToWork onto the goal stack.
We pop CanBikeToWork from the goal stack and look for rules that can be used to prove it. Rule d can be used to prove CanBikeToWork.
Rule d has the antecedents HaveBike, WorkCloseToHome, and Sunny. We push them onto the goal stack in reverse order: first Sunny, WorkCloseToHome, and HaveBike.
We first try rule e, which has the antecedent HaveMountainBike. This is a known fact, so we have proven that we have a bike.
We pop WorkCloseToHome from the goal stack.
We then move onto sunny weather. We know that it is raining, so the sunny conditions are not met. We can't bike to work.
Next, we try rule b, which has the antecedent CanDriveToWork. We can't use rule g, h, or i because we don't own a car.
We can try rule j, which has the antecedent CanRentCar. We push it on to the stack. Rule k can be used to prove CanRentCar. Rule k has 2 antecedents, which are HaveMoney and CarRentalOpen. We push these onto the stacks. HaveMoney is a known fact, so we can try rule l to prove CarRentalOpen. Rule l has HertzOpen as an antecedent. We know that HertzClosed based on known fact, so we are unable to use rule l. We now try to use rule m. Rule m has AvisOpen as an antecedent, so we push that onto the stack. Based on known fact, AvisOpen is satisfied, we pop that out of our stack, and since we HaveMoney as a known fact, and CarRentalOpen, we can pop those two out of the stack. After that, we can pop CanRentCar since rule k is now satisfied. Since we satisfied CanRentCar, we also satisfied CanDriveToWork(rule j). We can pop these out. We can then use rule b to prove CanGetToWork, and pop the goal CanGetToWork out of our stack.

Question 5:

When there is one starting point, and many solutions, it is better to use forward chaining. When there is one solution with many starting points, then it is better to use backward chaining. We can use forward chaining for tasks like medical diagnosis, planning systems, monitoring, controlling systems. We might want to use backward chaining for assisting proofs, game theory applications. Backward chaining usually starts with the steps that are closer to the solution, rather than starting at the beginning point.