Solutions for Assignment 3

1. a. Correct, \((A \land B)\) is true and only true when both A and B are true, then \((A \iff B)\) is also true.

b. Correct under \(A = T, B = F\).

c. Incorrect, if Smoke = F, Fire = T, we would then have \((\text{Smoke} \Rightarrow \text{Fire}) = T\), but \((\neg\text{Smoke} \Rightarrow \neg\text{Fire}) = F\), so it is satisfiable for this assignment.

d. Correct, right side: \(x=x\) is always true, the disjunction is true no matter what's on the left.

e. Incorrect. Statement is equivalent to \(\exists x \neg \text{play Football for}(x, \text{UO}) \lor \text{National Champion}(x)\). It is true for anyone who do not play football for the UO and it doesn't matter he gets national champion or not. It is so trivial to be true because most people in the world do not play football for the UO. It doesn't make sense to say it.

f. Incorrect: \(\forall x \neg(\neg\text{play Football for}(x, \text{UO}) \lor \neg\text{National Champion}(x))\)

It can be translated into: \(\forall x \text{ play Football for}(x, \text{UO}) \land \text{National Champion}(x)\). It means all people must play football for the UO AND get National Champion. It is so hard to be true and it doesn’t make sense either.

g. Correct, apply De Morgan's law, you get \(\exists x (P \land Q) \iff \forall x \neg(P \lor Q) \iff \forall x (\neg P \lor \neg Q)\)

2. Backward chaining is better.
Backward chaining starts with a list of goals (or a hypothesis) and works backwards from the consequent to the antecedent to see if there is data available that will support any of these consequents. While, Forward chaining starts with the data and reasons its way to the answer.

In this particular problem, the completely grounded KB (facts) is undefined and could be very large for all possible combinations from versions of parts. Backward chaining is better in time and space since we only care about these 10 parts.

3.

All Statements and Rules:

A: John is a student: \(\text{Student}(John)\)

B: AI is a CS course: \(\text{CS\_Course}(AI)\)

C: John takes AI class: \(\text{Take\_Course}(John, AI)\)
D: A student x finishes his degree: Finish_Degree(x)

E: Rule: AI Instructors always give good grade to the students. It means any student takes AI class can get good score.

∀x Take_Course (x, AI) ⇒ Good_Score(x, AI)

F: Rule: A student must get a good score in a CS course to finish a degree.

∀x, y Student (x) ∧ CS_Course (y) ∧ Take_Course (x, y) ∧ Good_Score(x, y) => Finish_Degree(x)

Backward chaining:

Hypothesis: Can John finish a degree? Finish_Degree (John) ?

Resolution:

First Change both Rule E and F to CNF:

E: ∀x ¬Take_Course (x, AI) ∨ Good_Score(x, AI)

F: ∀x, y ¬Student (x) ∨ ¬CS_Course (y) ∨ ¬Take_Course (x, y) ∨ ¬Good_Score(x, y) ∨ Finish_Degree(x)
Assume Finish_Degree (John) is false (¬Finish_Degree(John) is true)

Step 1: Resolve Rule F (in CNF) with ¬Finish_Degree(John) using {x/John}, we get

¬Student (John) ∨ ¬CS_Course (y) ∨ ¬Take_Course (John, y) ∨ ¬Good_Score(John, y) (1)

Step 2: Resolve (1) with Student (John), we get

¬CS_Course (y) ∨ ¬Take_Course (John, y) ∨ ¬Good_Score(John, y) (2)

Step 3: Resolve (2) with CS_Course (AI) using {y/Al}, we get

¬Take_Course (John, Al) ∨ ¬Good_Score(John, Al) (3)

Step 4: Resolve (3) with Take_Course(John, Al) , we get

¬Good_Score(John, Al) (4)

Step 5: Resolve Rule E with Take_Course(John, Al) using {x/John}, we get

Good_Score(John, Al) (5)

(4) and (5) are contradict to each other or they will be resolved to empty. So the assumption is false. Finish_Degree (John) is true.

Note, if you define Rule F as:

Student (x) ∧ CS_Course (y) ∧ Take_Course (x, y) ∧ Finish_Degree(x) ⇒ Good_Score(x, y)

by considering getting a good score in computer science course is a necessary condition but not a sufficient condition for getting a degree. You will not prove whether John can or cannot finish the degree. It keeps as an unknown based on the open world assumption (OWA). You will get full score if you show that both backward chaining and resolution will not prove Finish_Degree (John) is true or false.