Naman Jain Vimal Kumar
1001670153

Problem 1) Two logical statements A and B are logically equivalent if A <=> B. We have two knowledge bases, KB1 and KB2.. Write a function CHECK_EQUIVALENCE(KB1, KB2) that:

- ***Returns true if KB1 and KB2 are logically equivalent.***
- ***Returns false otherwise.***

Your pseudocode can use or modify any code from the textbook or slides, and can call any of the functions given in the textbook or slides, as long as such code and functions are used correctly, with correct names for the functions, and with well-specified values for all variables and arguments.

```
function CHECK_EQUIVALENCE (KB1, KB2) returns true or false
        inputs: KB1, the knowledge base 1 in propositional logic
                KB2, the knowledge base 2 in propositional logic

        symbols ← proposition symbols used in KB1 and KB2
        return COMPARE (KB1, KB2, symbols, {})

function COMPARE (KB1, KB2, symbols, model) returns true or false
        if EMPTY? (symbols) then
                if PLogic(KB1, model)
                then
                        return PLogic(KB2, model)
                else
                        return true
        else
                do
                P ← FIRST(symbols)
                rest ← REST(symbols)
                return (COMPARE (KB1, KB2, rest, model ∪ {P = true})
                        and COMPARE (KB1, KB2, rest, model ∪ {P = false})
                        and COMPARE (KB2, KB1, rest, model ∪ {P = true})
                        and COMPARE (KB2, KB1, rest, model ∪ {P = false}))
```

Note: PLogic() Returns true if a sentence holds within a model. The variable model represents a partial model—an assignment to some of the symbols.

Problem 2) KB and S1 are two propositional logic statements, that are constructed using symbols A, B, C, and using various connectives. The above truth table shows, for each combination of values of A, B, C, whether KB and S1 are true or false.

A	B	C	KB	S1
True	True	True	True	True
True	True	False	False	True
True	False	True	True	True
True	False	False	False	True
False	True	True	False	False
False	True	False	False	False
False	False	True	False	False
False	False	False	False	False

Part a: Given the above information, does KB entail S1? Justify your answer.

KB entails S1 only if all models of KB generates true value for S1. There are two models for KB that is D(A=True, B=True, C=True) and D(A=True, B=False, C=True). For the same data models, S1 also is True. Thus, we can say that KB entails S1.

Part b: Given the above information, does statement NOT(KB) entail statement NOT(S1)? Justify your answer.

Let us find the values of NOT(KB) and NOT(S1)

A	B	C	KB	NOT(KB)	S1	NOT(S1)
True	True	True	True	False	True	False
True	True	False	False	True	True	False
True	False	True	True	False	True	False
True	False	False	False	True	True	False
False	True	True	False	True	False	True
False	True	False	False	True	False	True
False	False	True	False	True	False	True
False	False	False	False	True	False	True

NOT(KB) entails NOT(S1) only if all models of NOT(KB) generates true value for NOT(S1). From the above table, we can see that

Problem 3) Suppose that some knowledge base contains various propositional-logic sentences that utilize symbols A, B, C, D (connected with various connectives). There are only two cases when the knowledge base is false:

- First case: when A is true, B is true, C is true, D is true.

- Second case: when A is true, B is false, C is true, D is false.

In all other cases, the knowledge base is true. Write a conjunctive normal form (CNF) for the knowledge base.

From the above information, we can construct the following truth table:

A	B	C	D	KB
True	True	True	True	False
True	True	True	False	True
True	True	False	True	True
True	True	False	False	True
True	False	True	True	True
True	False	True	False	False
True	False	False	True	True
True	False	False	False	True
False	True	True	True	True
False	True	True	False	True
False	True	False	True	True
False	True	False	False	True
False	False	True	True	True
False	False	True	False	True
False	False	False	True	True
False	False	False	False	True

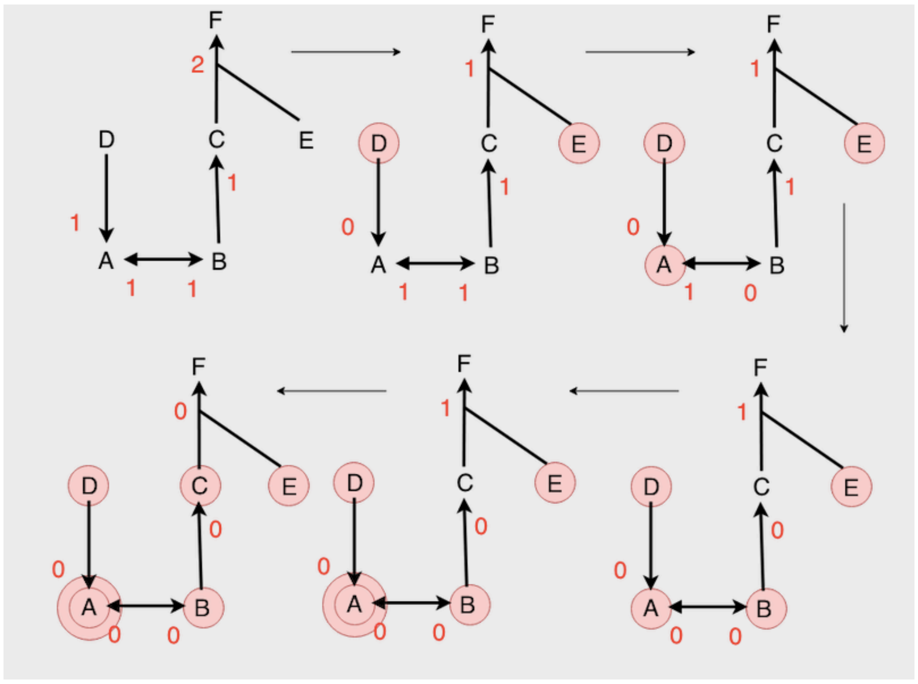
For the Knowledge Base, let ‘X’ be the conjunctive normal form.

$$X = \big((A \cap B \cap C \cap D) \cup (A \cap B^I \cap C \cap D^I)\big)^I$$
$$X = (A \cap B \cap C \cap D)^I \cap (A \cap B^I \cap C \cap D^I)^I$$
$$\boldsymbol{X = (A^I \cup B^I \cup C^I \cup D^I) \cap (A^I \cup B \cup C^I \cup D)}$$

Problem 4) Consider the KB

A <=> B
B => C
D => A
C AND E => F
E
D

Show that this entails C by
i. Forward Chaining



Initially all the graph is plotted based on the knowledge base. Based on the connections for each arrow, a value is assigned respectively. Connections are shown in the first diagram.

From the knowledge base, values that are true are circled red and the connections respective from them are reduced by 1. Here we reduce connections to ‘F’ by 1 and also ‘A’ by 1.

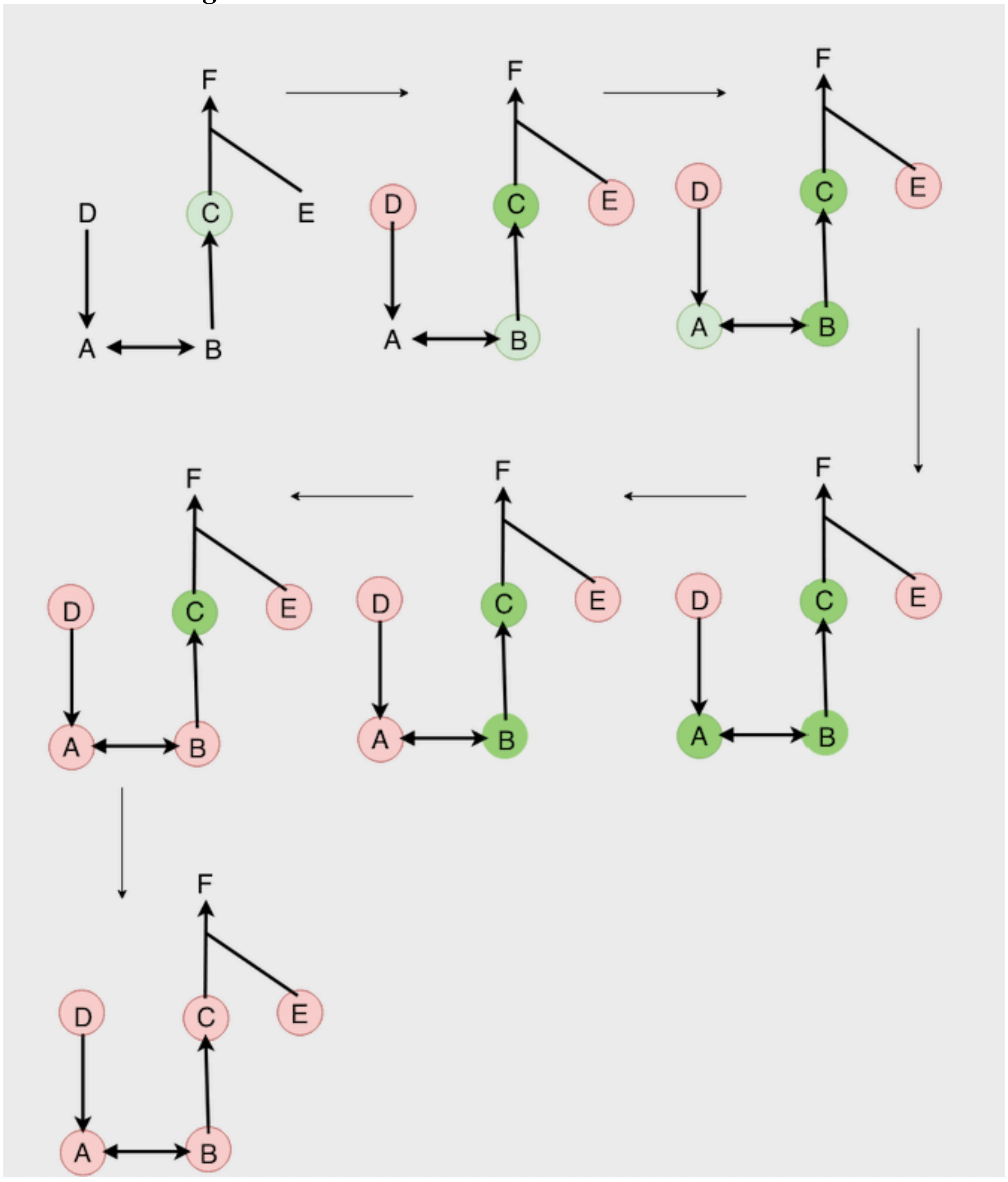
Among the connections, one with the connection equal to 0 is selected. Now connection to ‘A’ is 0, thus we select ‘A’. We change the state of ‘A’ to true (marked red) and reduce its connections by 1. Thus connection to ‘B’ is reduced to 0.

Now that connection to ‘B’ is 0, it is marked as true and its respective connections are updated. That is connection to ‘A’ is 0 and connection to ‘C’ is 0. The algorithm has 2 moves to make now, either ‘A’ or ‘C’. From the above diagram, we choose ‘A’ and revisit the state. There are no actions for the same.

Now ‘C’ is chosen and marked true, which is our final state. Thus using forward chaining, we can say that the above Knowledge Base entails ‘C’.

Order of States: { D, A, B, A, C }

ii. Backward Chaining



Here we start with our final state, that is ‘C’. We see states by which ‘C’ is connected to. From the knowledge base, ‘C’ is connected to ‘B’, thus ‘B’ is added to the stack. Similarly ‘A’ is connected to ‘B’ and is pushed to the stack and there by also ‘D’. ‘D’ is of true state, connection checking is stopped.

So currently our stack has [C, B, A, D]. Since 'D' is a true state, it is popped out. Now since 'A' is connected from 'D', which is a true state, even 'A' is marked true and popped out of the stack.

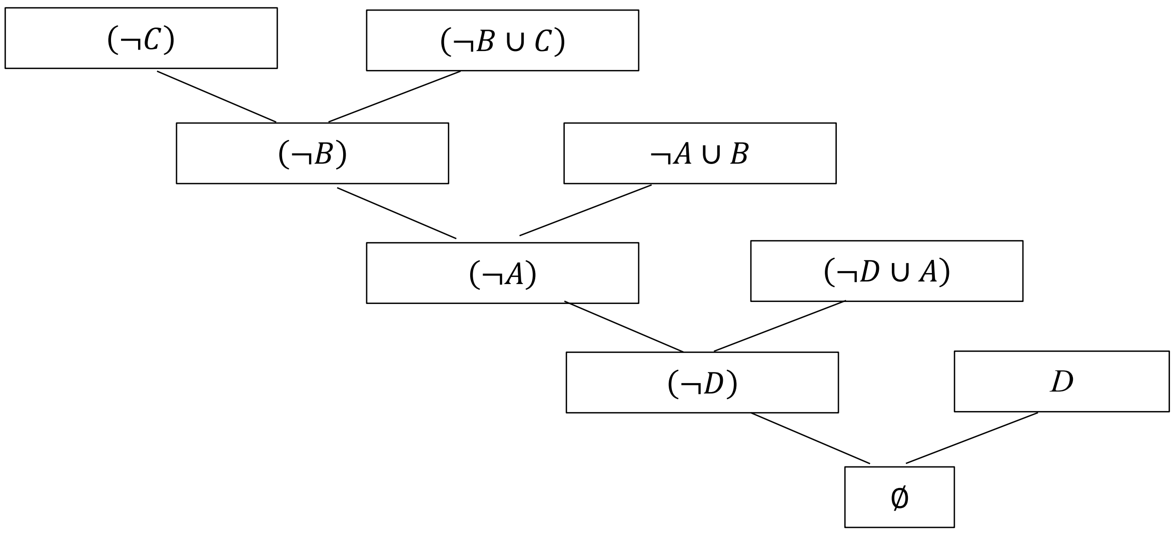
Top element of the stack 'B' is also connected from 'A' and has no other connections to it. Thus 'B' also is marked true and popped out. The only state left in the stack is 'C' and has no connections to it apart from 'B'. Thus 'C' is marked true and we could say that the knowledge base entails 'C'.

iii. Resolution

The above expressions should be reduced to CNF form

Equation	**Equivalent CNF**
A <=> B	= (A => B) ∩ (B => A) = $(\neg A \cup B) \cap (\neg B \cup A)$
B => C	= $(\neg B \cup C)$
D => A	= $(\neg D \cup A)$
C ∩ E => F	= (¬(C ∩ E) ∪ F) = $(\neg C \cup \neg E \cup F)$
D	= D
E	= E

Conclusion = C
Start With = $\neg C$



NULL achieved, thus knowledge base entails 'C'

Problem 5)
On April 20, 2017, John and Mary sign the following contract:
- ***If it rains on May 1, 2017, then John must give Mary a check for $10,000 on May 2, 2017***
- ***If John gives Mary a check for $10,000 on May 2, 2017, Mary must mow the lawn on May 3, 2017.***

What truly happened those days is the following:
- ***It did not rain on May 1, 2017***
- ***John gave Mary a check for $10,000 on May 2, 2017***
- ***Mary mowed the lawn on May 3, 2017.***

Let A = It rains on May 1, 2017,
B = John gives Mary a check for $10000 on May 2, 2017,
C = Mary moves the lawn on May 3, 2017

Part A: Write a propositional-logic statement to express the contract. Make sure that, for each symbol that you use, you clearly define what that symbol stands for.

$Contract = (A \rightarrow B) \cap (B \rightarrow C)$

Part B: Write a logical statement to express what truly happened. When possible, use the same symbols as in question 4a. If you need to define any new symbols, clearly define what those new symbols stand for.

$Actual = \neg A \cap B \cap C$

Part C: Was the contract violated or not, Justify your answer
The contract was not violated. As the event 'B' took place, that is John gave Mary a check worth $10000 on May 2, 2017, event 'C' also was completed by Mary to move the lawn on May 3, 2017. Event 'A' did not occur itself.

Problem 6) Consider a knowledge base with these facts:
- ***There is a dog called Shadow.***
- ***John gave Shadow to Mary.***
- ***If Shadow is male, Mary gave a smartphone to John.***
- ***If Shadow is female, Mary gave John a laptop.***
- ***John only gives male dogs to people.***
- ***Mary gave John a laptop.***

Convert the above knowledge-base to a first-order logic knowledge base. For each predicate, function, constant, or variable that you use, explicitly state:
- ***What type of entity it is (is it a predicate, function, constant, or variable).***
- ***What its semantics are (what it means).***

Knowledge Base	**First Order logic**
There is a dog named Shadow	$\exists a$, Dog(a) ∩ Name(a, Shadow)
John gave Shadow to Mary	Give(John, Shadow, Mary)

If Shadow is male, Mary gave a smartphone to John	isMale(Shadow) → Give(Mary, Smartphone, John)
If Shadow is female, Mary gave John a laptop	¬isMale(Shadow) → Give(Mary, Laptop, John)
John only gives male dogs to people	$\forall a\ \forall b,$ Give(John, Dog(a), Person(b)) → isMale(c)
Mary gave John a laptop	Give(Mary, Laptop, John)

Varibles: a, b, c
Constant: Shadow, John, Mary, Laptop, Smartphone
Predicates:

Name	***Semantic***
Dog(x)	To check if a given variable 'x' is a dog or not
Name(x, y)	To check if a name 'y' is assigned to the object 'x'
Give(x, y, z)	Affirmative if 'x' gives 'y' to 'z'
isMale(x)	To check if 'x' is male or not
Person(x)	To check if 'x' is a people or not

Functions: NULL

Problem 7)
Consider this first-order logic knowledge base:

```
        taller(John, Bill)
        (∀x) taller(x, Bill) => tall(x)
```

In this first-order logic knowledge base, taller and tall are predicates, x is a variable, and John, Bill are constants. Convert this first-order logic knowledge base into a propositional logic knowledge base, by performing the following two steps:

1. ***Define symbols for the propositional-logic version of the knowledge base, and specify what their equivalents are in the original first-order logic knowledge base.***
2. ***Define the statements that should be stored in the propositional-logic version of the knowledge base.***

The symbols you define should be comprehensive enough to allow us to translate any well-defined inference problem in the original knowledge base to an equivalent problem for the propositional knowledge base. Anything that we can infer from the original first-order logic knowledge base we should also be able to infer from the propositionalized knowledge base, and vice versa.

1.

taller(John, Bill)	***First Order Logic***
- Taller_John_Bill	taller(John, Bill)
- Taller_Bill_John	taller(Bill, John)
- Taller_John_John	taller(John, John)
- Taller_Bill_Bill	taller(Bill, Bill)

∀x taller(x, Bill)	***First Order Logic***
- Taller_x_Bill	taller(x, Bill)
- Taller_Bill_x	taller(Bill, x)
- Taller_x_x	taller(x, x)

Naman Jain Vimal Kumar
1001670153

- Taller_Bill_Bill	taller(Bill, Bill)
tall(x)	***First Order Logic***
- Tall_x	tall(x)
- Tall_Bill	tall(Bill)
- Tall_John	tall(John)

2.
Taller_John_Bill
Taller_x_Bill => Tall_x

Problem 8) Try and unifiy the following predicates(if possible)

```
taller(John, y), taller(x, Son(x))

taller(y, Barry), taller(Barry, x)

taller(x, Jane), taller(Bob, Jane)

taller(Son(x), Jane), taller(Bob, Jane)

taller(Barry, John), taller(x, y)
```

P	**Q**	***θ***
`taller(John, y)`	`taller(x, Son(x))`	{ x / John, y / Son(John) }
`taller(y, Barry)`	`taller(Barry, x)`	{ y / Barry, x / Barry }
`taller(x, Jane)`	`taller(Bob, Jane)`	{ x / Bob }
`taller(Son(x), Jane)`	`taller(Bob, Jane)`	Fail
`taller(Barry, John)`	`taller(x, y)`	{ x / Barry, y / John }