

CS 81, Logic and Computability

Problem Set 2: Natural Deduction for Propositional Logic

Problems 1–8 are natural deduction proofs. In Problems 1–6, you’ll use constructive logic. In Problems 7 and 8, you’ll use classical logic which allows the use of RAA (Reductio ad Absurdum), which is also called Proof by Contradiction or Indirect Proof (IP).

Please write your proofs for those problems using the online natural deduction proof checker at proofs.openlogicproject.org. The proof checker will show you that your proof is correct and give you a smiley face. Take a screenshot of the proof (e.g., using the Grab utility on a Mac or a similar tool) and submit a PDF version of the snapshot with the complete proof and the smiley face that shows that the proof checker verified the correctness of your proof.

Take a few minutes to look at the notation used by the `openlogicproject` online proof checker. The notation for the “Rules” are shown on the right side of the page (you only need the “Basic Rules”) and the notation for the operators (and, or, not, implication) are shown on the left side under “Instructions.” I suggest that you try to type in one or two proofs from class in this environment before you start trying your own new proofs. This environment takes just a bit of practice. I also strongly encourage you to do these sequent proofs *on paper* before you enter them into the proof checker. As we saw in class, it’s generally helpful to derive these proofs non-linearly, working backwards when necessary to ultimately get to our goal. It’s much easier to do that first on paper and then, afterwards, enter the proof online to be checked.

Here are a few things to keep in mind:

- When using AND or OR introduction, you don’t specify whether it’s left or right introduction. That makes things a bit easier!
- There are several different choices of symbols that you can use for the logic operators (e.g., and, or, not, implication).

Problem 1: [10 Points]

Use natural deduction for **constructive logic** in the `openlogicproject` to prove that: $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$.

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with "," or ";"):

$A \wedge (B \vee C)$

Conclusion:

$(A \wedge B) \vee (A \wedge C)$

CREATE PROBLEM

Proof:

Construct a proof for the argument: $A \wedge (B \vee C) \therefore (A \wedge B) \vee (A \wedge C)$

1	$A \wedge (B \vee C)$	
2	A	$\wedge E$ 1
3	$B \vee C$	$\wedge E$ 1
4	B	
5	$A \wedge B$	$\wedge I$ 2, 4
6	$(A \wedge B) \vee (A \wedge C)$	$\vee I$ 5
7	C	
8	$A \wedge C$	$\wedge I$ 2, 7
9	$(A \wedge B) \vee (A \wedge C)$	$\vee I$ 8
10	$(A \wedge B) \vee (A \wedge C)$	$\vee E$ 3, 4-6, 7-9

NEW LINE

NEW SUBPROOF

🎉 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 2: [10 Points]

Use natural deduction for constructive logic in the openlogicproject to prove that: $B \rightarrow A \vdash (A \rightarrow C) \rightarrow (B \rightarrow C)$.

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with ", " or ";"):

$B \rightarrow A$

Conclusion:

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

CREATE PROBLEM

Proof:

Construct a proof for the argument: $B \rightarrow A \therefore (A \rightarrow C) \rightarrow (B \rightarrow C)$

1	$B \rightarrow A$	
2	$A \rightarrow C$	
3	B	
4	A	$\rightarrow E$ 1, 3
5	C	$\rightarrow E$ 4, 2
6	$B \rightarrow C$	$\rightarrow I$ 3-5
7	$(A \rightarrow C) \rightarrow (B \rightarrow C)$	$\rightarrow I$ 2-6

NEW LINE

NEW SUBPROOF

😊 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 3: [10 Points]

Use natural deduction for constructive logic in the openlogicproject to prove that: $A \rightarrow \neg B \vdash B \rightarrow \neg A$.

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with "," or "and");

$A \rightarrow \neg B$

Conclusion:

$B \rightarrow \neg A$

CREATE PROBLEM

Proof:

Construct a proof for the argument: $A \rightarrow \neg B \therefore B \rightarrow \neg A$

1	$A \rightarrow \neg B$	
2	B	
3	A	
4	$\neg B$	$\rightarrow E\ 3, 1$
5	\perp	$\neg E\ 2, 4$
6	$\neg A$	$\neg I\ 3-5$
7	$B \rightarrow \neg A$	$\rightarrow I\ 2-6$

NEW LINE

NEW SUBPROOF

🎉 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 4: [10 Points]

Use natural deduction for constructive logic in the openlogicproject to prove that: $\neg\neg\neg A \vdash \neg A$. (Remember, this is constructive logic. We have no rule that eliminates a double negative!)

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with " " or " , "):

$\neg\neg A$

Conclusion:

$\neg A$

CREATE PROBLEM


Proof:

Construct a proof for the argument: $\neg\neg A \therefore \neg A$

1	$\neg\neg A$	
2	A	
3	$\neg A$	
4	\perp	$\neg E$ 2, 3
5	$\neg\neg A$	$\neg I$ 3-4
6	\perp	$\neg E$ 1, 5
7	$\neg A$	$\neg I$ 2-6

NEW LINE

NEW SUBPROOF

 Congratulations! This proof is correct.

Answer:

Problem 5: [10 Points]

Use natural deduction for constructive logic in the openlogicproject to prove that: $A \rightarrow \neg A \vdash \neg A$.

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with ", " or ";"):

$A \rightarrow \neg A$

Conclusion:

$\neg A$

CREATE PROBLEM

Proof:

Construct a proof for the argument: $A \rightarrow \neg A \therefore \neg A$

1	$A \rightarrow \neg A$	
2	A	
3	$\neg A$	$\rightarrow E$ 1, 2
4	\perp	$\neg E$ 2, 3
5	$\neg A$	$\neg I$ 2-4

NEW LINE

NEW SUBPROOF

😊 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 6: [10 Points]

Use natural deduction for constructive logic in the openlogicproject to prove that: $A \vee B \vdash \neg(\neg A \wedge \neg B)$.

Create a new problem

Click to edit

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with "," or ";"):

$A \vee B$

Conclusion:

$\neg(\neg A \wedge \neg B)$

CREATE PROBLEM

Proof:

Construct a proof for the argument: $A \vee B \therefore \neg(\neg A \wedge \neg B)$

1	$A \vee B$	
2	$(\neg A \wedge \neg B)$	
3	$\neg A$	$\wedge E$ 2
4	$\neg B$	$\wedge E$ 2
5	A	
6	\perp	$\neg E$ 3, 5
7	B	
8	\perp	$\neg E$, 4, 7
9	\perp	$\vee E$ 1, 5-6, 7-8
10	$\neg(\neg A \wedge \neg B)$	$\neg I$ 2-9

NEW LINE

NEW SUBPROOF

🎉 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 7: [10 Points]

Use natural deduction for classical logic in the openlogicproject to prove that: $\neg A \rightarrow A \vdash A$. (Note: Since this is in classical logic rather than constructive logic you may use RAA, called IP in openlogicproject, in your proof. In fact, you will need to use IP here.)

Create a new problem

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with "," or ";"):

$\neg A \rightarrow A$

Conclusion:

A

CREATE PROBLEM

Proof:

Construct a proof for the argument: $\neg A \rightarrow A \therefore A$

1	$\neg A \rightarrow A$	
2	$\neg A$	
3	A	$\rightarrow E$ 2, 1
4	\perp	$\neg E$ 2, 3
5	A	IP 2-4

NEW LINE

NEW SUBPROOF

😊 Congratulations! This proof is correct.

CHECK PROOF

START OVER

Answer:

Problem 8: [10 Points]

Use natural deduction for classical logic in the openlogicproject to prove that: $\neg(\neg A \wedge \neg B) \vdash A \vee B$. (Note: Since this is in classical logic rather than constructive logic you may use RAA, called IP in openlogicproject, in your proof. In fact, you will need to use IP here.)

Select if TFL or FOL syntax:

☒ TFL ☐ FOL

Premises (separate with "," or "and"):

$\neg(\neg A \wedge \neg B)$

Conclusion:

$A \vee B$

Proof:

Construct a proof for the argument: $\neg(\neg A \wedge \neg B) \therefore A \vee B$

1	$\neg(\neg A \wedge \neg B)$	
2	$\neg(A \vee B)$	
3	A	
4	$A \vee B$	$\vee I$ 3
5	\perp	$\neg E$ 2, 4
6	$\neg A$	$\neg I$ 3-5
7	B	
8	$A \vee B$	$\vee I$ 7
9	\perp	$\neg E$ 2, 8
10	$\neg B$	$\neg I$ 7-9
11	$\neg A \wedge \neg B$	$\wedge I$ 6, 10
12	\perp	$\neg E$ 1, 11
13	$(A \vee B)$	IP 2-12

😊 Congratulations! This proof is correct.

Answer:

Problem 9: Knights, Knaves, Gold! [10 Points]

You are on an island inhabited by knights and knaves. You may interact with the inhabitants of this island by only asking questions that can be answered with “yes” (true) or “no” (false). Knights always tell the truth and knaves always lie.

You arrive on the island believing that there may be gold to be found there. Almost immediately, you encounter an inhabitant. Your goal is to ask that inhabitant only one question and the answer to that question should tell you whether or not there is gold on the island.

1. What question should you ask? More precisely, what statement

would you present to the inhabitant such that their answer would reveal whether or not there is gold. For example, if you wanted to ask the question “Are you a knave and there is no gold on this island?” you would turn this into a statement “You are a knave and there is not gold on the island.”

2. Express your statement in propositional logic by defining a variable for each proposition in your statement and using those variables and logical connectives (\wedge , \vee , \neg , \rightarrow , \leftrightarrow) to express your statement. For example, if p = “you are a knave” and q = “there is gold on the island” then the statement “you are a knave and it is not the case that there is gold on the island” would be written as $p \wedge \neg q$.
3. Use a truth table to prove that your question allows you to determine whether or not there is gold on the island. Tip: You can add a column to your truth table showing the inhabitant’s response, and use that to show how the responses of the inhabitant allow you to determine precisely whether or not there is gold on the island. In other words, there is the *actual* truth value of your proposed logical statement, determined fully by the settings of the propositional variables, and then there is the response of the inhabitant, which might be different from the actual truth value of your logical statement, depending on whether they are a knight or a knave. You would then show that this response column allows you to determine with full accuracy whether there is gold on the island, no matter if the respondent is a knight or a knave. Also note that a knave does not negate propositions within your logical statement individually, but only negates the entire logical statement in the final step, as a whole.

Answer

1. The statement that one should use would be “there is gold if and only if you are a knight”.
2. Let p be the statement “there is gold” and let q be the statement “you are a knight”, we can see then that we can express the statement as $p \leftrightarrow q$.
3. We can construct the following truth table. It is important to note that that when $q = 0$, the statement that we are actually evaluating

is $\neg(p \leftrightarrow q)$ since we know that when $q = 0$, we have a knave and that knaves negate the entire statement. Therefore, keeping this in mind, we can construct the following table.

p	q	$p \leftrightarrow q$	$\neg(p \leftrightarrow q)$
0	0	1	0
1	0	0	1
0	1	0	1
1	1	1	0

We can see that considering our situation, the reality of our situation is reflected in our first two rows adjoined with the second two rows of our table, which we can see gives us truth values of 1 only when there is gold on the island.

Problem 10: Monks, Proofs, Transcendence! [10 Points]

A group of monks live in a monastery. They have taken a vow of silence, and cannot communicate with each other in any way. Each morning, the monks silently enter a large prayer chamber, where they all sit silently facing each other to pray, then return to their own rooms at night. They are expected to pray every day, but if they ever become aware that they have achieved transcendence, they will pack up their belongings in the night and leave the monastery. When a monk achieves transcendence, a luminous mark appears on their forehead. There are no mirrors in the monastery, so each monk cannot see their own forehead, though they do see the foreheads of all of the other monks when they are all in the prayer chamber together. One night (and on that night only!), which we'll call "night 0", while all the monks are in their own rooms, a booming voice rings out: "At least one of you has achieved transcendence!" The next day is denoted "day 1." You should assume that the monks are good at logic and will leave at the earliest time that it is possible for them to determine that they have transcended (and, of course, that a non-transcendent monk will not leave).

1. State a claim about what day the transcendent monks will leave if there are exactly n transcendent monks.

2. Prove your claim by induction.

Answer:

1. All the transcendent monks would leave on day n if there are exactly n transcendent monks.
2. Consider the following proof of the claim by induction.

Let us first consider the base case of where $n = 1$. We can see trivially that after night 0, the voice booms out and claims that at least one monk has achieved transcendence. However, as there is only one monk, it must follow that said monk is the one who has achieved transcendence, and thus, they leave on day 1. Thus, we have established our base case.

Now, let us assume that our inductive hypothesis holds for n monks. We will now prove that the statement holds for $n + 1$ monks. Consider that if we have $n + 1$ monks, we can consider that for days 1- n , each monk has insufficient information as to the number of monks who have achieved transcendence. That is, each sees n monks who have the marking for transcendence. Using our inductive case, we can see that for n monks, it takes n days for the monks to leave, but now, on the $n + 1^{th}$ date, each monk realizes that none of the n other monks have left, which indicates that they also see n monks with said marking. As a result, each monk realizes that they are indeed also marked and subsequently leave the monastery. Thus, our inductive hypothesis holds for $n + 1$ monks as desired.

Therefore, as our inductive proof holds for our base case and our inductive step, it follows that our proof holds for all n as desired.