CS50 Assignment 1 Notebook 2021 - Logical Thinking

October 3, 2021

1 Formal Analyses Assignment 1, Fall 2021

1.1 Logical Thinking

1.2 OVERVIEW

For this assignment, you will craft arguments both for and against a claim related to a Cornerstone "Big Question." You must then analyze the arguments in full, making strong applications of all of the logical thinking HCs. The arguments can be as simple or as sophisticated as you wish! Keep in mind that more complicated arguments will require a richer, more careful analysis. This may be a good way to showcase your knowledge, but could also lead to more flaws. Moreover, in pursuit of the appropriate difficulty level, you'll notice several "Optional" problems throughout the assignment to challenge yourself. These will only be scored if they are completed correctly with a thorough explanation. If you attempt an optional challenge but do not succeed, you will not be penalized with a low score.

1.3 FORMATTING & HC GUIDELINES

You must complete all tasks within this pre-formatted Jupyter notebook. **Please read and follow the formatting guidelines and the HC Guidelines in the assignment instructions on Forum (near the top and bottom of the instructions respectively).**

1.4 BIG QUESTION → ARGUABLE CLAIM

The main goal of the Cornerstone courses is to establish a foundation in the HCs that span the four core competencies: thinking critically, thinking creatively, interacting effectively, and communicating effectively. To help students learn the HCs and build practical knowledge, the curriculum is organized around a number of "Big Questions" that recur in multiple Cornerstones. The Big Questions provide some background content so that the HCs can be practiced and applied in a variety of relevant contexts. They also allow students to reflect on the applicability of the HCs in a larger context. For this assignment, you will argue two sides of a claim thematically related to a Big Question. You may choose any Big Question from the Cornerstone Curriculum Map that interests you. It does not matter whether the Big Question you choose has been introduced in your class yet or not. Next, develop an arguable stance on a component of the Big Question. **Please review ALL of the guidelines and examples provided to ensure your claim is arguable and sufficiently narrow/specific..**

1.4.1 Identify your chosen Big Question, your arguable claim (X), and a competing stance (Y).

Big Question: How is free choice manipulated? - X: Social media should not be regulated - Y: Social media should be regulated

1.4.2 Explain briefly why you chose this Big Question and how you identified a claim to argue for and against (<100 words).

I chose this Big Question and its arguments for its relevance, especially after episodes of prominent figures such as Donald Trump being banned from social media platforms. Also, it is common to debate about policies inside those platforms when crimes such as racism are often unpunished. I chose my two arguments based on discussions of free speech rights and how to work around that in social media regulation: should we regulate those platforms or not? Does regulation interfere with free choice?

1.5 PART 1

In this part, you will work with one side of the stance only (X). Craft two brief arguments in favor of this claim using deductive reasoning. One argument should be a valid deduction and one should contain a fallacy. Write these arguments in natural language in simple terms with clear use of logical connectives and atomic sentences so that it is amenable for the following analysis. Read the guidelines and examples in the assignment instructions for help crafting your arguments. [#deduction, #fallacies, #algorithms]

1.5.1 Write your stance and your deductive arguments.

X: Social media should not be regulated.

- P1: If people have free choice, people either use social media or don't.
- P2: People use social media.
- P3: If people use social media, then people want to share their ideas.
- P4: If people want to share their ideas, then people need a free space.
- P5: If people need a free space, then social media should not be regulated.
- X: If people have free choice, then social media should not be regulated.

- P1: If people have free choice, then people can communicate freely.
- P2: If people can communicate freely, then all spaces are free to communicate.
- P3: If all spaces are not free to communicate, then social media is not a free space.
- P4: If social media is not a free space, then social media should not be regulated.
- X: If people have free choice, then social media should not be regulated.

1.5.2 1. Clearly identify the features of these arguments that make them deductive and explain why (<75 words).

1. Rule of necessity: it's clear that the argument's conclusion follows directly from the premises. The way the arguments were constructed makes it impossible for the conclusion to be false when the premisses are true;
2. Conclusion: the conclusion "social media should not be regulated" is directly related to all premisses from the first one, "people have free choice";

3. Definition and rules VS observation and evidence: we can divide the premises into atomic sentences and use them as rules/definitions. They're not simply observations/evidence (particular to induction).
4. Form and specific words: The form of one premise following the other closely, the presence of connectors, and the absence of expressions such as "likely," "probable" indicate those are deductions.

1.5.3 2. For the valid deduction:

2.1 Translate your argument into symbolic logic and provide brief commentary for how you developed the translation.

- F: People have free choice.
- U: People use social media.
- I: People want to share their ideas.
- S: People need free space.
- R: Social media should be regulated.

Argument:

- $F \implies U \vee \neg U$
- U
- $U \implies I$
- $I \implies S$
- $S \implies \neg R$
- X: $F \implies \neg R$

The translation was done following the connectors on the natural language sentences. They have (somewhat) established meanings in symbolic language. In this case, the only atomic sentence that could confuse is the stance itself: "Social media should NOT be regulated." Sentences with words that indicate negative values such as "not" should be carefully analyzed and translated. Here, the atomic sentence "R" represents the negation of the stance, but using this translation makes it easier for the logician not to confuse the value of the sentence if negations come into play. - The combination of "if… then" indicates conditionals here - "or" indicates disjunction - "not" and "don't" indicates negation

2.2 Choose two sentences from the argument and negate them using DeMorgan's laws. Ideally, you should choose two sentences that have different logical connectives. Translate the negations back into English. First sentence: - $F \implies U \vee \neg U$

Negation using DeMorgan's laws:

1. Re-write the sentence as:
    - $\neg F \vee (U \vee \neg U)$
2. Negate the sentence:
- \$ $\neg(\neg F \vee (U \vee \neg U))$\$
3. Apply DeMorgan's laws:
    - \$ $\neg\neg F \wedge \neg(U \vee \neg U)$\$\$$\neg\neg F \wedge \neg U \wedge \neg\neg U$\$
- \$ F $\wedge \neg U \wedge U$\$
4. Translate back into English:
    - "People have free choice and can use social media and can't use social media"

Second sentence: - $S \implies \neg R$

Negation using DeMorgan's laws:

1. Re-write the sentence as:
   - $\neg S \lor \neg R$
2. Negate the sentence:
   - $\neg(\neg S \lor \neg R)$
3. Apply DeMorgan's laws:
   - $\neg\neg S \land \neg\neg R)$
   - $S \land R$
4. Translate back into English:
   - "People need free space and social media should be regulated"

2.3 Prove that the argument is indeed valid using a formal proof. Premisses: 1. P1: $F \implies U \lor \neg U$ 2. P2: U 3. P3: $U \implies I$ 4. P4: $I \implies S$ 5. P5: $S \implies \neg R$

Prove:

6. **F** (assumption)
7. $U \lor \neg U$ (Modus Ponens - 1, 6)
8. **U** (Disjunctive Sylogism - 2)
9. **I** (Modus Ponens - 3, 8)
10. **S** (Modus Ponens - 4, 9)
11. **$** $\neg R\$(ModusPonens - 5, 10)$

2.4 Discuss the soundness of the argument (<100 words). First of all, it's important to notice people indeed either use social media or don't use it; this is a tautology. We can argue that people who use social media usually do it to share their ideas, but they're not limited to it. People can use those platforms to gather information, read the news, and not share their own ideas. Also, a "free space" is not necessarily linked to a place where anything of any sort can be said or done. For people to have free choice, they need to have safe spaces to share their ideas. However, free choice does not include racism, xenophobia, etc. Gaps in argumentation like these could weaken the conclusion.

2.5 *Optional*: Negate each of the premises of your argument and derive a valid conclusion, being sure to explain the derivation and comment on the soundness. If it is impossible to form a valid argument from these premises, explain why.

1.5.4 Original Argument:

1. $F \implies U \lor \neg U$
2. U
3. $U \implies I$
4. $I \implies S$
5. $S \implies \neg R$
6. X: $F \implies \neg R$

1.5.5 Negations:

Re-writing premisses:

1. $\neg F \vee (U \vee \neg U)$
2. U
3. $\neg U \vee I$
4. $\neg I \vee S$
5. $\neg S \vee \neg R$
6. X: $\neg F \vee \neg R$

Negating:

1. $\neg(\neg F \vee (U \vee \neg U))$
2. $\neg U$
3. $\neg(\neg U \vee I)$
4. $\neg(\neg I \vee S)$
5. $\neg(\neg S \vee \neg R)$
6. X: $\neg(\neg F \vee \neg R)$

**Using DeMorgan's Laws*:* 1. $F \wedge \neg U \wedge U$ 2. $\neg U$ 3. $U \wedge \neg I$ 4. $I \wedge \neg S$ 5. $S \wedge R$ 6. X: $F \wedge R$

Derivated arguments using the atomic sentences:

Original sentences: - F: People have free choice. - U: People use social media. - I: People want to share their ideas. - S: People need free space. - R: Social media should be regulated.

Translations:

1. People have free choice and people don't use social media and people use social media.
2. People don't use social media.
3. People use social media and people don't want to share their ideas.
4. People want to share their ideas and people don't need free space.
5. People need free space and social media should be regulated.
6. X: People have free choice and social media should be regulated.

Analyzing how the translations turned out to be, one premise doesn't follow the other. Also, the "combination" of the atomic sentences don't make much sense: "People want to share their ideas, and people don't need free space," when in reality, we do know people need free space - they can't share their ideas in highly regulated territories such as dictatorships. We have a problem with soundness here. The premises are bad ones. But if we focus on the conclusion, we managed to arrive at argument Y, defined at the beginning of the assignment "Social media should be regulated," which is the exact opposite of argument X: "Social media should not be regulated." We end up having a valid argument (shown on the truth-table bellow), but not a sound one. Also, it isn't elementary to prove its validity using formal proofs since they don't relate directly to conjunctions.

1.5.6 Truth-Table

```
[1]: def premise1(f, u):
         # this function takes into consideration what premise 1 means in logic␣
      ↪language
```

```
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return f and not u and u

def premise2 (u):
    # this function takes into consideration what premise 2 means in logic␣
↪language
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return not u

def premise3(u, i):
    # this function takes into consideration what premise 3 means in logic␣
↪language
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return u and not i

def premise4(i, s):
    # this function takes into consideration what premise 4 means in logic␣
↪language
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return i and not s

def premise5(s, r):
    # this function takes into consideration what premise 5 means in logic␣
↪language
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return s and r

def conclusion(f, r):
    # this function takes into consideration what the conclusion means in logic␣
↪language
    # it inputs the atomic sentences and outputs the truth-values of the␣
↪sentence
    return f and r

def valid_checker(p1, p2, p3, p4, p5, concl):
    # this function tests if the whole argument is valid and shows a truth table
    print('f\t u\t i\t s\t r\t p1\t p2\t p3\t p4\t p5\t conclusion')
    # prints the 1st row of the truth table

    validity = True
```

```
    # here we test all combinations between the atomic sentences and its truth␣
↪values
    for f in [True,False]:
        for u in [True,False]:
            for i in [True,False]:
                for s in [True, False]:
                    for r in [True, False]:

                        # here we print all atomic sentence's truth values in a␣
↪truth table
                        print(f,'\t',u,'\t',i,'\t', s,'\t', r,'\t',p1(f,␣
↪u),'\t', p2(u), '\t', \
                        p3(u, i),'\t',p4(i, s), p5(s, r),'\t',concl(f, r))

                        # the 'validity' value just changes to False if all␣
↪premises are
                        # True and the conclusion is False
                        # this is a clear indication if the argumentation was␣
↪did correctly
                        # a valid deductive argument follows such a logical␣
↪structure that it's impossible
                        # for its premisses to be True and the conclusion False
                        if p1(f, u)==True and p2(u)==True and p3(u, i)==True and␣
↪p4(i, s)==True and p5(s, r)==True and concl(f, r)==False:
                            validity = False

    # if the argument is valid, we'll print one especific string
    # if the argument is not valid, we'll print another string
    if validity == True:
        print("The argument is VALID.")
    else:
        print("The argument is INVALID.")

# calls the valid_checker function
valid_checker(premise1, premise2, premise3, premise4, premise5, conclusion)
```

```
f        u        i        s        r        p1       p2       p3       p4       p5
conclusion
True     True     True     True     True     False    False    False    False True
True
True     True     True     True     False    False    False    False    False False
False
True     True     True     False    True     False    False    False    True False
True
True     True     True     False    False    False    False    False    True False
False
True     True     False    True     True     False    False    True     False True
```

```
True
True     True    False   True    False   False   False   True    False False
False
True     True    False   False   True    False   False   True    False False
True
True     True    False   False   False   False   False   True    False False
False
True     False   True    True    True    False   True    False   False True
True
True     False   True    True    False   False   True    False   False False
False
True     False   True    False   True    False   True    False   True False
True
True     False   True    False   False   False   True    False   True False
False
True     False   False   True    True    False   True    False   False True
True
True     False   False   True    False   False   True    False   False False
False
True     False   False   False   True    False   True    False   False False
True
True     False   False   False   False   False   True    False   False False
False
False    True    True    True    True    False   False   False   False True
False
False    True    True    True    False   False   False   False   False False
False
False    True    True    False   True    False   False   False   True False
False
False    True    True    False   False   False   False   False   True False
False
False    True    False   True    True    False   False   True    False True
False
False    True    False   True    False   False   False   True    False False
False
False    True    False   False   True    False   False   True    False False
False
False    True    False   False   False   False   False   True    False False
False
False    False   True    True    True    False   True    False   False True
False
False    False   True    True    False   False   True    False   False False
False
False    False   True    False   True    False   True    False   True False
False
False    False   True    False   False   False   True    False   True False
False
False    False   False   True    True    False   True    False   False True
```

```
False
False    False    False    True     False    False    True     False    False False
False
False    False    False    False    True     False    True     False    False False
False
False    False    False    False    False    False    True     False    False False
False
The argument is VALID.
```

1.5.7 3. Python

3.1 Use Python to assess the validity of the argument. The code should output whether or not the argument is valid (it should be valid!) based on a truth table. You are not required to print the full truth table, but it might provide you with a helpful sanity check. You may use a similar code from session 5 (3.1), simply adapting it to your argument, or if you wish, write your own from scratch. The code from lesson 4 (2.2) may also be helpful. Whatever approach you take, you must clearly explain how the code works (as described below).

As with any code, you need to include clearly annotated in-line comments to document what the code is doing and how it is using logic. Read this resource about the importance of comments and this one for further guidance.

In addition to your brief in-line comments, you must also include an overarching explanation of what the code is doing as a whole in connection with #deduction. In particular, you should explain how the truth table is used to ascertain validity and how this logic concept is incorporated in the code (<150 words).

[2]:
```python
def conditional(a,b):
    # inputs two Boolean variables representing atomic sentences a, b
    # outputs the truth value of the sentence "a -> b"
    return not a or b

def extra_premise (u):
    # inputs u and always returns True as a value (tautology)
    return u or not u

def premise1(f, u):
    # takes premise 1 atomic sentences and inserts in conditional function to␣
 ↪test its truth values
    # here we use the function extra_premise as a parameter since our condition␣
 ↪in the 1st premise
    # takes U or not U as a sentence letter
    return conditional(f, extra_premise(u))

def premise2(u, i):
    # takes premise 2 atomic sentences and inserts in conditional function to␣
 ↪test its truth values
    return conditional(u, i)
```

```python
def premise3(i, s):
    # takes premise 3 atomic sentences and inserts in conditional function to␣
 ↪test its truth values
    return conditional(i, s)

def premise4(s, r):
    # takes premise 4 atomic sentences and inserts in conditional function to␣
 ↪test its truth values
    return conditional(s, not r)

def conclusion(f, r):
    # takes the conclusion atomic sentences and inserts in conditional function␣
 ↪to test its truth values
    # here we can see Modus Ponens in action
    # (we use the 1st atomic sentence and achieve the last - our conclusion)
    return conditional(f, not r)

def valid_checker(p1, ext, p2, p3, p4, concl):
    # this function tests if the whole argument is valid and shows a truth table
    print('f\t u\t i\t s\t r\t p1\t ext\t p2\t p3\t p4\t conclusion') # prints␣
 ↪the 1st row of the truth table

    validity = True

    # here we test all combinations between the atomic sentences and its truth␣
 ↪values
    for f in [True,False]:
        for u in [True,False]:
            for i in [True,False]:
                for s in [True, False]:
                    for r in [True, False]:

                        # here we print all atomic sentence's truth values in a␣
 ↪truth table
                        print(f,'\t',u,'\t',i,'\t', s,'\t', r,'\t',p1(f,␣
 ↪u),'\t', ext(u), '\t', \
                        p2(u, i),'\t',p3(i, s), p4(s, r),'\t',concl(f, r))

                        # the 'validity' value just changes to False if all␣
 ↪premises are True and the conclusion is False
                        # this is a clear indication if the argumentation was␣
 ↪did correctly
                        # a valid deductive argument follows such a logical␣
 ↪structure that it's impossible
                        # for its premisses to be True and the conclusion False
```

```
                        if p1(f, u)==True and ext==True and p2(u, i)==True and␣
↪p3(i, s)==True and p4(s, r)==True and concl(f, r)==False:
                            validity = False

    # if the argument is valid, we'll print one especific string
    # if the argument is not valid, we'll print another string
    if validity == True:
        print("The argument is VALID.")
    else:
        print("The argument is INVALID.")

# calls the valid_checker function
valid_checker(premise1,extra_premise, premise2,premise3,premise4, conclusion)
```

```
f       u       i       s       r       p1      ext     p2      p3      p4
conclusion
True    True    True    True    True    True    True    True    True False
False
True    True    True    True    False   True    True    True    True True
True
True    True    True    False   True    True    True    True    False True
False
True    True    True    False   False   True    True    True    False True
True
True    True    False   True    True    True    True    False   True False
False
True    True    False   True    False   True    True    False   True True
True
True    True    False   False   True    True    True    False   True True
False
True    True    False   False   False   True    True    False   True True
True
True    False   True    True    True    True    True    True    True False
False
True    False   True    True    False   True    True    True    True True
True
True    False   True    False   True    True    True    True    False True
False
True    False   True    False   False   True    True    True    False True
True
True    False   False   True    True    True    True    True    True False
False
True    False   False   True    False   True    True    True    True True
True
True    False   False   False   True    True    True    True    True True
False
True    False   False   False   False   True    True    True    True True
```

```
True
False    True    True    True    True    True    True    True    True False
True
False    True    True    True    False   True    True    True    True True
True
False    True    True    False   True    True    True    True    False True
True
False    True    True    False   False   True    True    True    False True
True
False    True    False   True    True    True    True    False   True False
True
False    True    False   True    False   True    True    False   True True
True
False    True    False   False   True    True    True    False   True True
True
False    True    False   False   False   True    True    False   True True
True
False    False   True    True    True    True    True    True    True False
True
False    False   True    True    False   True    True    True    True True
True
False    False   True    False   True    True    True    True    False True
True
False    False   True    False   False   True    True    True    False True
True
False    False   False   True    True    True    True    True    True False
True
False    False   False   True    False   True    True    True    True True
True
False    False   False   False   True    True    True    True    True True
True
False    False   False   False   False   True    True    True    True True
True
The argument is VALID.
```

I adapted session's 4 code to fit my needs. The 'conditional' function is beneficial because of the repetitive conditions in my argumentation. The function makes me not repeat multiple times the act of analyzing a conditional statement.

The 'premises functions' take into account the differences in types of conditionals and the different atomic sentences each deals with. Here we can't have just one function to do everything; each premise is one of its own.

The validity_checker function prints a truth table with all the arguments' truth-values and gives us the information if the whole argument is valid or not.

The 'for loops' access the possible values each atomic sentence can assume. Those values enter the 'print function'.

Then the 'premises functions' are called. They return the truth-values they can assume. Those values are a combination of the atomic sentences' truth-values in the context of each premise construction, e.g.: p2(u, i) returns 'not u or i', this means p2 return all truth-values combinations of the sentence 'not u or i'. 'not u or i' takes 'u' and 'i' possible truth-values to output its own boolean.

The validity_checker function changes the value of the variable 'validity' if there's invalidity in the argument. The invalidity is made visual in a truth table when all premises are 'True,' and the conclusion is 'False.' For an argument to be invalid, we need one of the possible truth-value combinations to respect this form.

3.2 ***Optional challenge*****: Write your own Python code that will create a truth table for any logical statement involving any number of atomic sentences and connectives. You can decide on the input/output format, but explain in detail what you are doing and why. In particular, be sure to explain why your work constitutes an effective application of #algorithms.**

[3]:
```
# write your code for 3.2 (optional) here
```

3.3 ***Optional super challenge*****: Write your own Python code that will check the validity of any set of logical sentences that form an argument. Explain in detail how your code works. You can decide on the input/output format, but explain in detail what you are doing and why. Additionally, be sure to comment on why your work constitutes an effective application of #algorithms**

[4]:
```
# write your code for 3.3 (optional) here
```

1.5.8 4. For the fallacious argument (<200 words):

4.1 Name the type of fallacy or fallacies involved and justify the fallacious nature of the argument by explaining the flaw in logic.

4.2 Explain one way to mitigate this fallacy and describe the effect of the mitigation on the meaning of the conclusion and/or premises.

(Translating this part of the argument into symbolic logic is not required but could be useful for your explanations.)

- P1: If people have free choice, then people can communicate freely.
- P2: If people can communicate freely, then all spaces are free to communicate.
- P3: If all spaces are not free to communicate, then social media is not a free space.
- P4: If social media is not a free space, then social media should not be regulated.
- X: If people have free choice, then social media should not be regulated.

Translation to Symbolic Language

- F: People have free choice.
- C: People can communicate freely.
- S: All spaces are free to communicate.
- M: Social media is a free space.
- R: Social media should be regulated.

Sentences: - P1: $F \implies C$ - P2: $C \implies S$ - P3: $\neg S \implies \neg M$ - P4: $\neg M \implies \neg R$ - X: $F \implies \neg R$

Fallacies: Translating the argumento into sybolic language makes it clear where the fallacy is located. Since we are dealing with a deductive argument, the fallacy is a formal one and indicated in the format of the argument. It is noticeable that from P2 to P3 we commit a Non Sequitur Of Relevance fallacy. That means that the conclusion in P3 (M) does not follow the other premises. When we affirm $\neg S$ the only thing we can affirm is $\neg C$. For us to correctly affirm $\neg M$ we would need to have an extra premise:

- $C \implies S$
- $\neg S \implies \neg C$
- $\neg C \implies \neg M$. (extra premise)

When we affirm "If all spaces are not free to communicate, then social media is not a free space" we should be able to conclude that "social media is not a free space" because people can't communicate freely, but there isn't any premisse or logical idea that connects both. Negating S only give us $ $\neg C\$ automatically, but not a connection between \neg C$ and $\neg M$.

We can then fix this argument adding a new premise:

- P1: If people have free choice, then people can communicate freely.
- P2: If people can communicate freely, then all spaces are free to communicate.
- P3: If all spaces are not free to communicate, then people can't communicate freely.
- EXTRA: If people can't communicate freely, then social media is not a free space.
- P4: If social media is not a free space, then social media should not be regulated.
- X: If people have free choice, then social media should not be regulated.

Even if we logically fix the argument, it still turns to be unsound. In the real world, the connection between not being able to communicate freely and social media not being a free space is not clear. One can argue social media is the place where you can actually be free and speak your mind. This would imply that even if the premise "If people can't communicate freely" is true, "then social media is not a free space" is not necessarily true, which is a clear case of an invalid argument.

Going further, we can evaluate this argument using a truth-table and conclude this is an INVALID argument:

[5]:
```
def conditional(a,b):
    # inputs two Boolean variables representing atomic sentences a, b
    # outputs the truth value of the sentence "a -> b"
    return not a or b

def premise1(f, c):
    # takes premise 1 atomic sentences and inserts in conditional function to␣
↪test its truth values
    return conditional(f, c)

def premise2(c, s):
    # takes premise 2 atomic sentences and inserts in conditional function to␣
↪test its truth values
    return conditional(c, s)
```

```python
def premise3(s, m):
    # takes premise 3 atomic sentences and inserts in conditional function to␣
↪test its truth values
    return conditional(not s, not m)

def premise4(m, r):
    # takes premise 4 atomic sentences and inserts in conditional function to␣
↪test its truth values
    return conditional(not m, not r)

def conclusion(f, r):
    # takes the conclusion atomic sentences and inserts in conditional function␣
↪to test its truth values
    return conditional(f, not r)

def valid_checker(p1, p2, p3, p4, concl):
    # this function tests if the whole argument is valid and shows a truth table
    print('f\t c\t s\t m\t r\t p1\t p2\t p3\t p4\t conclusion') # prints the 1st␣
↪row of the truth table

    validity = True

    # here we test all combinations between the atomic sentences and its truth␣
↪values
    for f in [True,False]:
        for c in [True,False]:
            for s in [True,False]:
                for m in [True, False]:
                    for r in [True, False]:

                        # here we print all atomic sentence's truth values in a␣
↪truth table
                        print(f,'\t',c,'\t',s,'\t', m,'\t', r,'\t',p1(f,␣
↪c),'\t', \
                        p2(c, s),'\t',p3(s, m), p4(m, r),'\t',concl(f, r))

                        # the 'validity' value just changes to False if all␣
↪premises are True and the conclusion is False
                        # this is a clear indication if the argumentation was␣
↪did correctly
                        # a valid deductive argument follows such a logical␣
↪structure that it's impossible
                        # for its premisses to be True and the conclusion False
                        if p1(f, c)==True and p2(c, s)==True and p3(s, m)==True␣
↪and p4(m, r)==True and concl(f, r)==False:
                            validity = False
```

```python
    # if the argument is valid, we'll print one especific string
    # if the argument is not valid, we'll print another string
    if validity == True:
        print("The argument is VALID.")
    else:
        print("The argument is INVALID.")

# calls the valid_checker function
valid_checker(premise1, premise2,premise3,premise4, conclusion)
```

```
f       c       s       m       r       p1      p2      p3      p4
conclusion
True    True    True    True    True    True    True    True True       False
True    True    True    True    False   True    True    True True       True
True    True    True    False   True    True    True    True False      False
True    True    True    False   False   True    True    True True       True
True    True    False   True    True    True    False   False True      False
True    True    False   True    False   True    False   False True      True
True    True    False   False   True    True    False   True False      False
True    True    False   False   False   True    False   True True       True
True    False   True    True    True    False   True    True True       False
True    False   True    True    False   False   True    True True       True
True    False   True    False   True    False   True    True False      False
True    False   True    False   False   False   True    True True       True
True    False   False   True    True    False   True    False True      False
True    False   False   True    False   False   True    False True      True
True    False   False   False   True    False   True    True False      False
True    False   False   False   False   False   True    True True       True
False   True    True    True    True    True    True    True True       True
False   True    True    True    False   True    True    True True       True
False   True    True    False   True    True    True    True False      True
False   True    True    False   False   True    True    True True       True
False   True    False   True    True    True    False   False True      True
False   True    False   True    False   True    False   False True      True
False   True    False   False   True    True    False   True False      True
False   True    False   False   False   True    False   True True       True
False   False   True    True    True    True    True    True True       True
False   False   True    True    False   True    True    True True       True
False   False   True    False   True    True    True    True False      True
False   False   True    False   False   True    True    True True       True
False   False   False   True    True    True    True    False True      True
False   False   False   True    False   True    True    False True      True
False   False   False   False   True    True    True    True False      True
False   False   False   False   False   True    True    True True       True
The argument is INVALID.
```

1.6 PART 2

For the competing stance (Y), craft two brief arguments using inductive reasoning (please see the guidance provided above for how to construct your arguments from your chosen question). One argument should be a strong induction and one should contain a fallacy. [#induction, #fallacies] Big Question: How is free choice manipulated? - Y: Social media should be regulated

Strong induction: - P1: Dipayan Ghosh (Harvard Business Review, 2021) reported that social media is highly used to spread misleading content. Politicians like Donald Trump used these platforms to inflame followers for years - in his case, the posts ended up inciting the Capitol invasion in January 2021. - P2: Because social media algorithms work to keep users scrolling, they tend to amplify conspiracy theories and are generally considered inappropriate content. - P3: As reported by Janice Asare (Forbes, 2021), because of its algorithm, social media many times miss-ban posts such as anti-racist ones. - P4: The algorithm gives companies money, so it is not their priority to work on better solutions and environment to detect what exactly is hate speech and things to be removed from their platforms (Asare J., Forbes, 2021). - P5: The level of "freedom" and no punishment inside social media causes extremists to be comfortable speaking their minds. This creates an environment of insecurity for minorities.

- Y: Therefore, social media should be regulated.

Fallacious induction: - P1: Social media should be regulated. If this doesn't happen, more hate speech will be shared. - P2: If more hate speech is shared, more people will be incited to act considering hate as a good reason. - P3: If we have more incited people, we'll probably have more events like the Capitol invasion repeating itself.

- Y: Therefore, if we want to prevent scenes like the Capitol invasion, we should regulate social media.

1.6.1 1. Clearly identify the features of these arguments that make them inductive and explain why (<75 words).

In this case, for us to reach a conclusion, we use data and observation. Here we're dealing with pieces of evidence that may lead to a reasonable conclusion. We don't have a relationship of necessity. Even if the conclusion is strong and related to the premises, we don't have a relationship such as in deductive arguments that if the premises are all True, the conclusion has to be True too. The presence of words such as "indicates," "probably," "tendency" clearly shows the inexistence of a relationship of necessity.

1.6.2 2. For the strong induction (<200 words):

2.1 Is this a specific type of induction? Which type, and why?

2.2 Elaborate on the strength and reliability of the argument. Explain how the argument could be made more reliable (including by increasing the strength). Causal Inference. We can observe that hate speech causes an insecure environment, and to avoid that, we need rules for what social media users can say. We can then infer that regulations on these spaces that address this specific problem will minimize inappropriate content. The argument has a good level of strength. It is reasonable to argue that since social media is full of harmful content, we should limit those as much as possible through regulations. We could make the argument stronger, showcasing successful cases of rules that created better spaces (through an analogy, prediction, or even authority evaluations). One argument against regulations is the fear of disrespecting free speech, so to make the argument stronger, it's interesting to define free speech and its limitations - hate speech being one. Using a specific definition makes it easier for us to work on arguments that respect it and avoid counterarguments that explore lacky premises.

1.6.3 3. For the fallacious argument (<200 words):

3.1 Name the type of fallacy or fallacies involved and justify the fallacious nature of the argument by explaining the flaw in logic.

3.2 Explain one way to mitigate this fallacy and describe the effect of the mitigation on the meaning of the conclusion and/or premises.

- P1: Social media should be regulated. If this doesn't happen, more hate speech will be shared.
- P2: If more hate speech is shared, more people will be incited to act considering hate as a good reason.
- P3: If we have more incited people, we'll probably have more events like the Capitol invasion repeating itself.
- Y: Therefore, if we want to prevent scenes like the Capitol invasion, we should regulate social media.

The central fallacy in the argument is the 'slippery slope.' Considering the conditional nature of the premises, progression of events is possible but still avoidable. The sequence in the premises creates a considerable distance between the first premise and the conclusion. It is weak to affirm that the first premise will make a domino effect and cause the conclusion. In conditionals, we usually use the idea of one thing 'possibly' causing another. Here we work with the concept of necessity: If A, then 'necessarily' B. One way to mitigate this fallacy is to rephrase premises such as "Social media should be regulated. If this doesn't happen, more hate speech will be shared." to "Social media should be regulated. If this doesn't happen, more hate speech will likely be shared." By doing this, we can break the effect of 'eternal progression.' We can also make our conclusion without linking it to a specific one-time situation; there are better inferences we can make.

1.7 REFLECTION

1.7.1 Write a reflection about how you approached this assignment and what you learned. Address the following points. [#scienceoflearning]

- While working on this assignment, how did the principles from the science of learning deepen your knowledge of #induction, #deduction, #algorithms, and #fallacies while working on this assignment? (<100 words)
- Throughout both sections, what did you learn about the connection between formal logic, language, computer code, and the use of logical thinking in the real world? (<100 words)

Deep processing, desirable difficulty, and deliberate practice were essential to write the assignment. Deep processing helped me to understand the HC's closely and continuously come back to them. The process of interacting with the concepts made me refine my arguments and explanations. Desirable difficulty made me have fun with the assignment. It was a challenge but doable with time and deliberate practice - getting help from my peers taught me a lot. Throughout the assignment's sections, I realized how wording my premises interfered with my final product. "Social media should be regulated" is not the same as "Social media should be more regulated." It was also interesting to see how some things make sense logically, but they are confusing in natural language. The process of negating premises was particularly revealing in this sense - the final argument ended up being logically valid, but it wasn't sound. I realized the usage of formal logic, combined with coding, is incredibly useful to evaluate real arguments made by other people. We can divide premises into atomic sentences and create logic sentences that the computer can understand and assess the conclusion's validity. This gets harder with induction, but we can spot fallacies and inconsistencies when evaluating specific words and connections between premises. Becoming skilled in logical thinking goes beyond mathematics and logic itself; it's about effective and truthful communication.

1.7.2 *Optional reflection questions*

These will only be scored if the application makes a sufficiently strong connection with an HC, constituting "deep" (4-level) knowledge, so please tag an HC using the guidelines in the full assignment instructions in Forum. Besides the logical thinking HCs targeted on this assignment, you might consider #critique for the first question and #gapanalysis for the second.

Optional **1: What side of the claim do you believe was argued more effectively and why? Consider applying #critique. (<200 words)** → Write your explanation here for the first optional reflection question.

Optional **2: Create a new assignment problem for future classes that you think is a good test of the skills covered by one or more of these HCs. Provide the complete step-by-step solution, and a justification for why this is a good assignment problem. It should be entirely different from this assignment (consider applying #gapanalysis).** ***The FA team would like to thank Dima Lkhagvatogtokh (M22) and Silen Naihin (M24) for their submissions to this question in 2018 and 2020; together, their ideas formed the basis for the assignment that you just completed!*** → Write your explanation here for the second optional reflection question.

YOU'RE DONE! You must upload TWO files:

1. Primary: A **PDF** of your entire assignment. A PDF of your entire assignment. This is to be submitted as a separate file, NOT simply inside the zipped folder. Email attachments will not be accepted. We encourage students to follow the tips available in this guide, *especially the best practices listed at the end.*

2. Secondary: A **zipped folder** containing the .ipynb file and any other relevant files for running the notebook.

```
[ ]:
```

```
[ ]:
```