305 Lecture 07 - Direct Derivations

Brian Weatherson

July 8, 2020

Basic Idea

A derivation is a series of steps that get you from the premises to the conclusion, with every step falling into one of a small number of approved kinds of transition.

Justification

The big thought is that no step could take you from truth to falsity (or to non-truth).

- So you can string as many of these steps together as you like, and it will never take you from truth to falsity.
- And to justify the procedure, you just need to justify the various kinds that are allowed.

Example

1. P
2. $\neg\neg P \rightarrow \neg\neg Q$

Q

Intuitive Argument

- Start assuming that the two premises are true.

Intuitive Argument

- Start assuming that the two premises are true.
- If P is true, then $\neg\neg P$ is true.

Intuitive Argument

- Start assuming that the two premises are true.
- If P is true, then $\neg\neg P$ is true.
- So $\neg\neg P$ is true.

Intuitive Argument

- Start assuming that the two premises are true.
- If P is true, then $\neg\neg P$ is true.
- So $\neg\neg P$ is true.
- If $\neg\neg P$ and $\neg\neg P \to \neg\neg Q$ are true, then $\neg\neg Q$ is true.

Intuitive Argument

- Start assuming that the two premises are true.
- If P is true, then $\neg\neg P$ is true.
- So $\neg\neg P$ is true.
- If $\neg\neg P$ and $\neg\neg P \to \neg\neg Q$ are true, then $\neg\neg Q$ is true.
- So $\neg\neg Q$ is true.

Intuitive Argument

- Start assuming that the two premises are true.
- If P is true, then $\neg\neg P$ is true.
- So $\neg\neg P$ is true.
- If $\neg\neg P$ and $\neg\neg P \to \neg\neg Q$ are true, then $\neg\neg Q$ is true.
- So $\neg\neg Q$ is true.
- And that implies Q is true, as required.

Formal Argument in Carnap

```
1. Show: Q
2.      P          :PR
3.      ~~P -> ~~Q :PR
4.      ~~P        :DNI 2
5.      ~~Q        :MP 4, 3
6.      Q          :DNE 5
7. :DD 6
```

Carnap

```
1. Show: Q
2.      P             :PR
3.      ~~P -> ~~Q    :PR
4.      ~~P           :DNI 2
5.      ~~Q           :MP 4, 3
6.      Q             :DNE 5
7. :DD 6
```

- A lot of what I'm going to say over the next few slides is about **Carnap**, not about logic in general.

Natural Deduction

```
1. Show: Q
2.      P             :PR
3.      ~~P -> ~~Q    :PR
4.      ~~P           :DNI 2
5.      ~~Q           :MP 4, 3
6.      Q             :DNE 5
7. :DD 6
```

- This is a version of what is known as a **natural deduction** proof system.
- It is somewhat non-standard, but that's not to say any one way is standard.

Natural Deduction

```
1. Show: Q
2.     P             :PR
3.     ~~P -> ~~Q    :PR
4.     ~~P           :DNI 2
5.     ~~Q           :MP 4, 3
6.     Q             :DNE 5
7. :DD 6
```

- What is common to all natural deduction systems is that when you read the steps, they read like a (pedantic version of) ordinary language reasoning.

Starting and Ending

```
1. Show: Q
2.      P            :PR
3.      ~~P -> ~~Q   :PR
4.      ~~P          :DNI 2
5.      ~~Q          :MP 4, 3
6.      Q            :DNE 5
7. :DD 6
```

- The most idiosyncratic feature of Carnap is the first and last line of the derivation.

Starting

```
1. Show: Q
2.     P             :PR
3.     ~~P -> ~~Q :PR
4.     ~~P           :DNI 2
5.     ~~Q           :MP 4, 3
6.     Q             :DNE 5
7. :DD 6
```

- In Carnap, you have to start a proof by announcing where you are headed.

Ending

```
1. Show: Q
2.       P              :PR
3.       ~~P -> ~~Q :PR
4.       ~~P            :DNI 2
5.       ~~Q            :MP 4, 3
6.       Q              :DNE 5
7. :DD 6
```

- And you end the proof by saying which line it is that the conclusion is reached.

Starting and Ending

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q   :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

- Note that these are the only two lines that are not indented.
- Proof systems (Carnap included) are visual, graphic systems, and vertical and horizontal arrangements tend to have meaning.

Starting and Ending

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q   :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

- They are also the only lines here that do not have a justification.
- Those abbreviations and numbers to the right of the other lines are justifications - you don't include them on the start or the finish.

Ending

```
1. Show: Q
2.      P            :PR
3.      ~~P -> ~~Q   :PR
4.      ~~P          :DNI 2
5.      ~~Q          :MP 4, 3
6.      Q            :DNE 5
7. :DD 6
```

- The ‘DD’ at the end is to indicate this is a **direct** derivation.
- We’ll get to the contrast with indirect derivations presently.

Premises

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q   :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

After the introductory line, the first lines are the premises - if they exist.

Premises

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

The premises need to be noted - that's what the 'PR' is for - but they are not derived.

Premises

```
1. Show: Q
2.      P              :PR
3.      ~~P -> ~~Q :PR
4.      ~~P            :DNI 2
5.      ~~Q            :MP 4, 3
6.      Q              :DNE 5
7. :DD 6
```

Your justification for writing them is that they are the beginning of what you are trying to prove.

Premises

```
1. Show: Q
2.      P              :PR
3.      ~~P -> ~~Q :PR
4.      ~~P            :DNI 2
5.      ~~Q            :MP 4, 3
6.      Q              :DNE 5
7. :DD 6
```

So they don’t get line numbers afterwards.

Derived Lines

```
1. Show: Q
2.      P             :PR
3.      ~~P -> ~~Q    :PR
4.      ~~P           :DNI 2
5.      ~~Q           :MP 4, 3
6.      Q             :DNE 5
7. :DD 6
```

- From now on, every line will be derived from previous lines.
- And the justification for it will be a rule, plus some line or lines.

Derived Lines

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

In Carnap the premises and derived lines are indented.

- The indenting is **four spaces**. For reasons I don't understand, a tab character here won't work.

Double Negation Introduction

```
1. Show: Q
2.     P           :PR
3.     ~~P -> ~~Q  :PR
4.     ~~P         :DNI 2
5.     ~~Q         :MP 4, 3
6.     Q           :DNE 5
7. :DD 6
```

If φ is a line, then you can add $\neg\neg\varphi$ as a new line.

Double Negation Introduction

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q   :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

The rule that you are using is abbreviated to 'DNI', and you have to justify this by citing the line where φ appears.

Double Negation Introduction

```
1. Show: ~~~~P
2.     P            :PR
3.     ~~P          :DNI 2
4.     ~~~~P        :DNI 3
5. :DD 4
```

This isn't specific to DNI, but note that for any rule, the input lines can be either a premise or a derived line.

- The rules do not distinguish between premises and derived lines.

Rules

A rule says that given sentences of some form, another particular sentence can be written.

Rules

To apply the rule correctly, you have to do 3 things

1. The sentence has to be the right one given the constraints of the rule.
2. You have to write down (immediately after a colon) the abbreviation for the rule.
3. You have to write down the line, or lines, that provide the inputs.

Double Negation Introduction

```
1. Show: Q
2.      P             :PR
3.      ~~P -> ~~Q :PR
4.      ~~P           :DNI 2
5.      ~~Q           :MP 4, 3
6.      Q             :DNE 5
7. :DD 6
```

Line 4 is allowed because you can add ¬¬ to any line by the rule Double Negation Introduction.

Double Negation Introduction

```
1. Show: Q
2.     P             :PR
3.     ~~P -> ~~Q    :PR
4.     ~~P           :DNI 2
5.     ~~Q           :MP 4, 3
6.     Q             :DNE 5
7. :DD 6
```

The abbreviation for Double Negation Introduction is DNI - so that's what we write.

Double Negation Introduction

```
1. Show: Q
2.     P            :PR
3.     ~~P -> ~~Q   :PR
4.     ~~P          :DNI 2
5.     ~~Q          :MP 4, 3
6.     Q            :DNE 5
7. :DD 6
```

And the input, the line we are adding $\neg\neg$ to, is line 2, so we write ‘2’.

A Trap

This is not a good proof - why not?

```
1. Show: ~~P -> Q
2.     P -> Q      :PR
3.     ~~P -> Q    :DNI 2
4. :DD 3
```

A Trap

You have to add the negations to **the whole sentence**.

- So the correct output here is $\neg\neg(P \to Q)$

Modus Ponens

```
1. Show: Q
2.      P            :PR
3.      ~~P -> ~~Q :PR
4.      ~~P          :DNI 2
5.      ~~Q          :MP 4, 3
6.      Q            :DNE 5
7. :DD 6
```

- The rule at line 5 is the most important in this part of the course.
- It even gets a fancy Latin name.

Modus Ponens

Given inputs $\varphi \to \psi$ and φ, infer ψ

Modus Ponens

- The abbreviation is MP.
- The line numbers are the lines where $\varphi \to \psi$ and φ appear.

Line Numbers

- There is a detail that some people get confused by at this point.
- The line numbers are the lines where the immediate inputs to the rule come from.
- They don't list all the justifications for those lines.
- So we list line 4, because it is where $\neg\neg P$ is, but not line 2, from where we derived line 2
- At every stage, we are just looking at whether that immediate step is ok.

A Trap

- As with DNI, it is important to apply the rule only to whole sentences.
- The sentence $\varphi \to \psi$ has to have $\to$ as its **main connective**.

Modus Ponens

This is OK.

```
1. Show: Q \/ R
2.     P -> (Q \/ R) :PR
3.     P             :PR
4.     Q \/ R        :MP 2, 3
5. :DD 4
```

This is **not** OK.

```
1. Show: Q \/ R
2.     (P -> Q) \/ R :PR
3.     P             :PR
4.     Q \/ R        :MP 2, 3
5. :DD 4
```

Modus Tollens

There is another rule that I haven't included in the example proof - modus tollens.

- It takes as input a line saying $\varphi \to \psi$, and a line saying $\neg\psi$.
- And it outputs a line saying $\neg\varphi$.

Differences between MP and MT

Different input

- In MP, the input is the left hand side, the **antecedent** of the conditional.
- In MT, the input is the **negation** of the **right hand side**, or **consequent** of the conditional.

Different output

- In MP, the output is the right hand side, the **consequent** of the conditional.
- In MT, the output is the **negation** of the **left hand side** of the conditional.

Double Negation Elimination

- This rule takes as input a sentence of the form $\neg\neg\varphi$.
- And it returns as output the sentence φ.

Double Negation Elimination

```
1. Show: Q
2.     P           :PR
3.     ~~P -> ~~Q  :PR
4.     ~~P         :DNI 2
5.     ~~Q         :MP 4, 3
6.     Q           :DNE 5
7. :DD 6
```

- The abbreviation is DNE.
- And because there is only one input, there is only one line cited.

That’s All!

```
1. Show: Q
2.      P             :PR
3.      ~~P -> ~~Q :PR
4.      ~~P           :DNI 2
5.      ~~Q           :MP 4, 3
6.      Q             :DNE 5
7. :DD 6
```

Since the line matches what was to be shown, we have a complete ‘direct derivation’.

Four Rules

Modus Ponens (MP) From $\varphi \to \psi$ and φ, infer ψ

Modus Tollens (MT) From $\varphi \to \psi$ and $\neg\psi$, infer $\neg\varphi$

Double Negation Introduction (DNI) From φ, infer $\neg\neg\varphi$

Double Negation Elimination (DNE) From $\neg\neg\varphi$, infer φ

Restrictions and Things to Remember

- Apply the negations in DNI to the whole sentence.
- Make sure the arrow is the main connective for MP and MT
- Cite the lines where the ‘from’ sentences appear in the proof.

Carnap is fussy about spacing

- Four spaces for the indented sentences.
- No space ever after a colon.
- One space after the abbreviation for the rule.
- These are not part of ‘logic’ in any sense - they are rules for this particular computer program.