

Logic, First Course, Winter 2020. Week 7, Lecture 1. [Back to course website](#)

Negation

In this section, we introduce how negation is handled in natural deduction, and illustrate it by doing several examples.

- [The falsum symbol](#)
- [Elimination rule for negation](#)
- [Introduction rule for negation](#)
- [Connection between arrow and falsum and negation](#)
- [A very common pattern](#)
- [Illustrating the common pattern](#)
- [Illustrating the common pattern again](#)
- [A more challenging example](#)

The falsum symbol

To describe the introduction and elimination rules for negation, we need to introduce a special symbol. This symbol is \perp and is called falsum or bottom or bot. Intuitively, it's a special symbol for a contradiction, something that's always false.

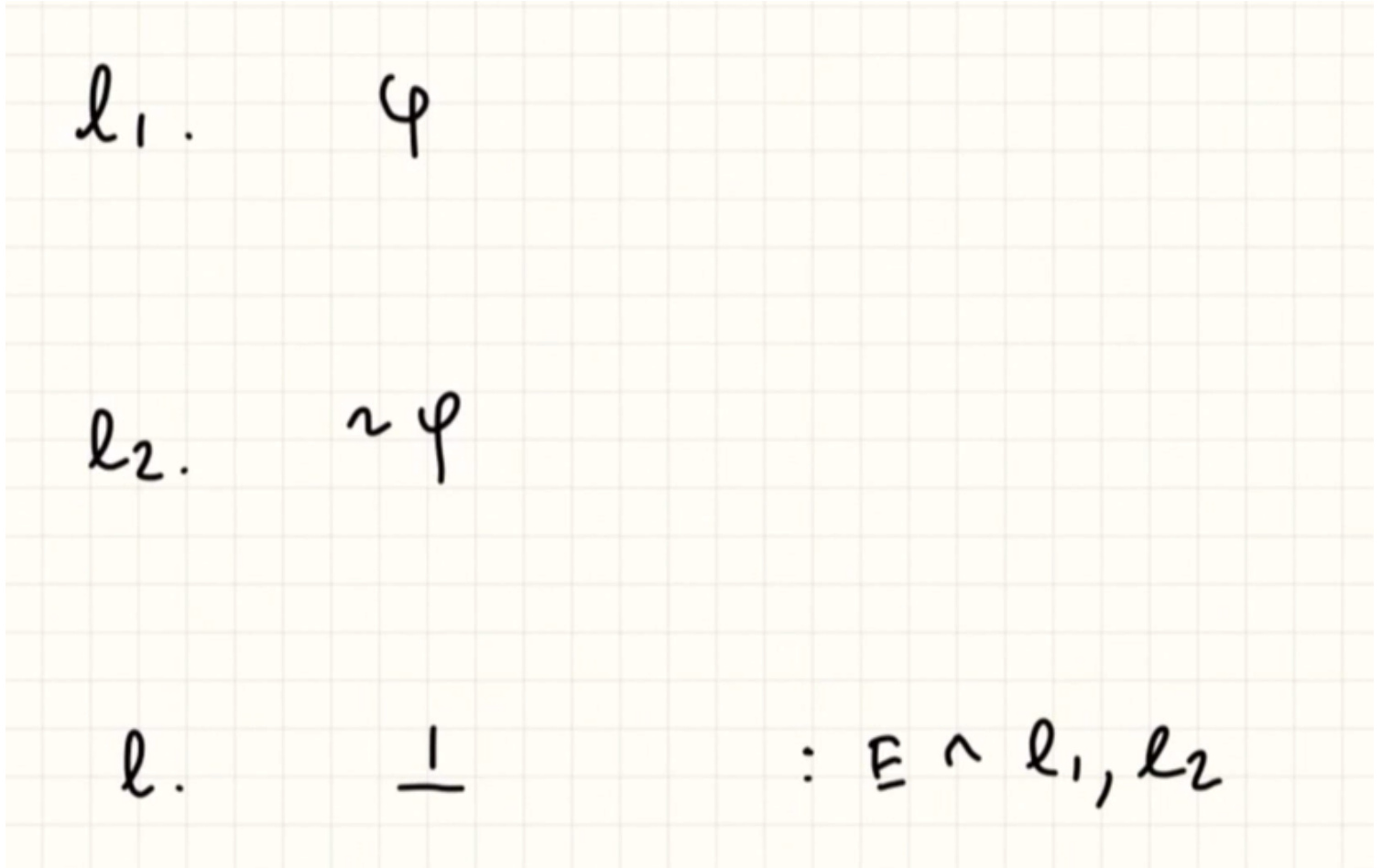
Initially it may seem strange to have one symbol which stands for all contradictions. However, contradictions validly imply anything. Hence, any two contradictions are equivalent. Thus if we wanted our deductive system to match up with our judgements of validity, perhaps it is plausible that we have one symbol \perp which stands for all contradictions.

The falsum symbol \perp can be typed in one of three ways. First, you can just cut and paste from this screen or anywhere else. Second, you can alternatively type `!?`. Third, you can program your computer to accept a shortcut like `\bot` or anything else you want. (Here are external sites which contain instructions on how to do this on a [Mac](#) or [Windows](#)).

Elimination rule for negation

Here is the elimination rule for negation: if you have ϕ on line ℓ_1 and you have $\neg\phi$ on line ℓ_2 , then you may write \perp on any subsequent line $\ell > \ell_1, \ell_2$.

Here is a picture of the rule:



In this rule, it does not matter whether the line with ϕ comes before or after the line with $\neg\phi$.

Example 1. $p \rightarrow \neg q, q, p \vdash \perp$.

First we do it by hand:

Example 1 $p \rightarrow \neg q, q, p \vdash \perp$

1. $p \rightarrow \neg q$: assumption
2. q : assumption
3. p : assumption
4. $\neg q$: $E \rightarrow 1, 3$
5. \perp : $E \neg 2, 4$

Second, you can try to input it into the proof-checker yourself, or come back later and practice:

exercise

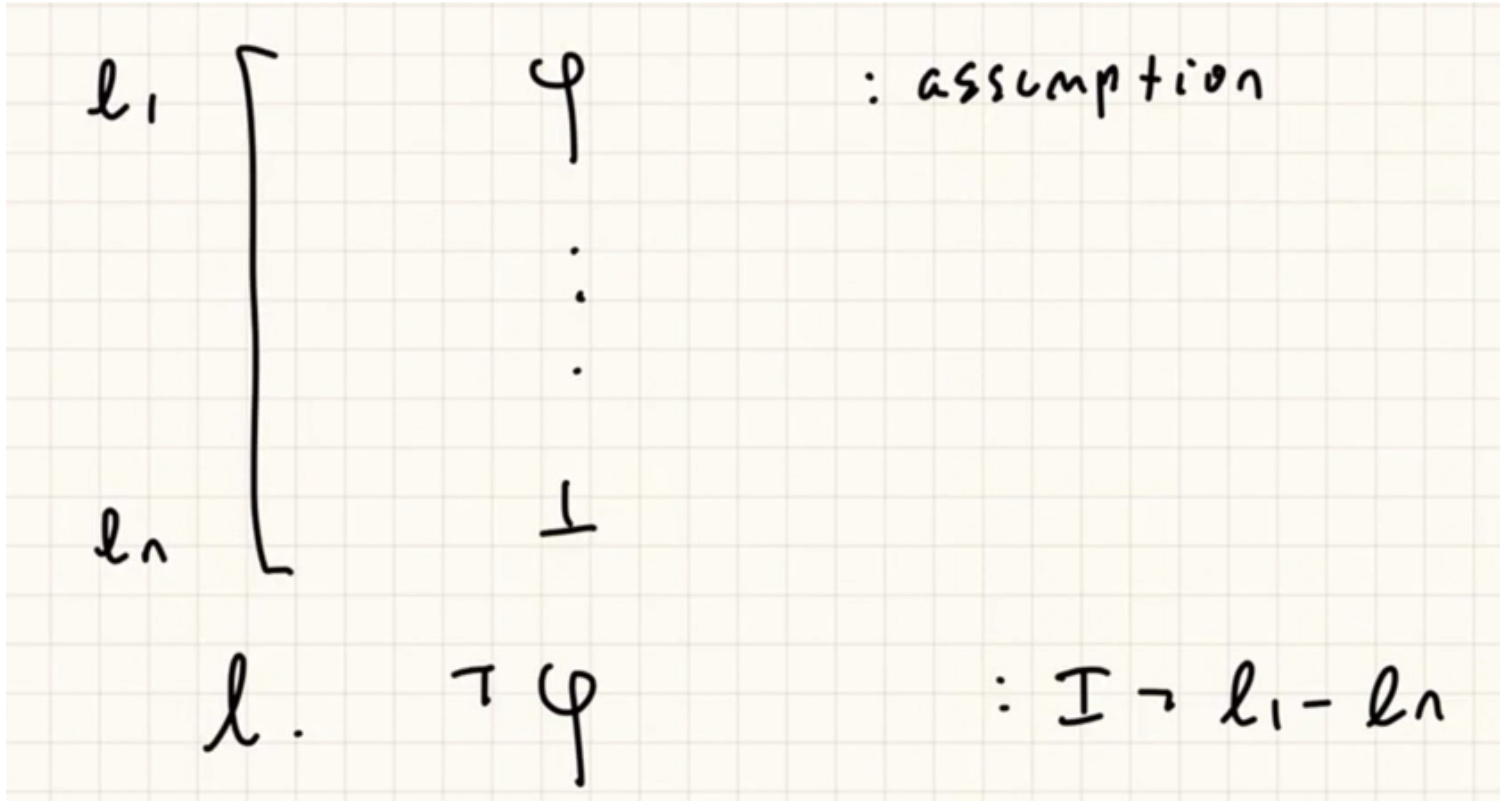
$(p \rightarrow \neg q), q, p \vdash \perp$

1. $p \rightarrow \neg q$: assumption
2. q : assumption
3. p : assumption

Introduction rule for negation

Here is the introduction rule for negation: suppose that consecutive lines $\ell_1 - \ell_n$ constitute a bracketed proof with premise ϕ and conclusion \perp . Then one may introduce $\neg\phi$ at any subsequent line $\ell > \ell_n$.

Here is a picture of the rule:



Example 2. $p \vdash \neg\neg p$.

First we do it by hand:

Example 2

$$p \vdash \neg\neg p$$

1. p : assumption

2. $\neg p$: assumption

3. \perp : $E\neg$ 1, 2

4. $\neg\neg p$: $I\neg$ 2-3

Second, you can try to input it into the proof-checker yourself, or come back later and practice:

exercise

$p \vdash \neg\neg p$

1. p : assumption

Connection between arrow and falsum and negation

The introduction rule for negation looks a lot like [the introduction rule for arrow](#) which we learned previously. In particular, to introduce $p \rightarrow q$ you just need a proof inside a bracket which starts at p and ends at q . To introduce $\neg p$, you just need a proof inside a bracket which starts at p and ends at \perp . Hence, one way to think about all this is as follows: negations are abbreviations for arrow statements, and in particular $\neg p$ is an abbreviation for $p \rightarrow \perp$. However, since we use it all the time, it's convenient to take negation as a primitive symbol.

A very common pattern

A very common pattern to get used to and to anticipate in these proofs is this one:

1.	ψ	: assumption
[2.	ψ : assumption
	.	
	.	
	.	
	50.	$\neg \psi$
[51.	\perp : $E \neg 1, 50$
	52.	$\neg \psi$: $I \neg 2 - 51$

Basically, when you want to prove a negation, you need a bracket with a falsum at the bottom. But the only way to get a falsum is to get ϕ and $\neg\phi$ for some ϕ .

Illustrating the common pattern

Example 3. $q \vdash \neg(p \wedge \neg q)$.

First we do it by hand:

Example 3

$$q \vdash \neg(p \wedge \neg q)$$

1. q : assumption
2. $\left[\begin{array}{l} p \wedge \neg q \end{array} \right. : \text{assumption}$
3. $\left[\begin{array}{l} \neg q \end{array} \right. : \wedge E \ 2$
4. $\left[\begin{array}{l} \perp \end{array} \right. : E \neg \ 1, \ 3$
5. $\neg(p \wedge \neg q) : I \neg \ 2 - 4$

Second, you can try to input it into the proof-checker yourself, or come back later and practice:

exercise

$$q \vdash \neg(p \wedge \neg q)$$

1. q : assumption

Illustrating the common pattern again

Here's another example

Example 4. $q, p \rightarrow \neg q \vdash \neg p$.

First we do it by hand:

Example 4 $q, p \rightarrow \neg q \vdash \neg p$

1. q : assumption
2. $p \rightarrow \neg q$: assumption
3. $\left[\begin{array}{l} p \\ \neg q \\ \bot \end{array} \right.$: assumption
4. $\neg q$: $E \rightarrow 2, 3$
5. \bot : $E \neg 3, 4$
6. $\neg p$: $I \neg 3-5$

Second, you can try to input it into the proof-checker yourself, or come back later and practice:

exercise

$q, (p \rightarrow \neg q) \vdash \neg p$

1. q :assumption
2. $p \rightarrow \neg q$:assumption

A more challenging example

Here a final example:

Example 5. $\neg\neg\neg p \vdash \neg p$.

First we do it by hand:

Example 5 : $\sim \sim \sim p \vdash \sim p$

1. $\sim \sim \sim p$: assumption

2. p : assumption

3. $\sim p$: assumption

4. \perp : $E \sim \underline{2}, \underline{3}$

5. $\sim \sim p$: $I \sim \underline{3-4}$

6. \perp : $E \sim \underline{1}, \underline{5}$

7. $\sim p$: $I \sim \underline{2-6}$

Second, you can try to input it into the proof-checker yourself, or come back later and practice:

exercise

$$\neg\neg\neg p \vdash \neg p$$

1. $\neg\neg\neg p$:assumption

This example shows that triple negation implies single negation. By using *Example 2* one can show that the converse holds as well. In later lectures, we will look at the associated results about [the law of double-negation](#).

These are lecture notes written for [this course](#).¹

1. It is run on the Carnap software, which is [↩](#)

An **Open Tower** project. Copyright 2015-2019 G. Leach-Krouse <gleachkr@ksu.edu> and J. Ehrlich