Natural deduction for propositional logic

Michael Franke

Semantic vs. syntactic approach to logical inference; derivation; derivation rule; soundness & completeness; natural deduction system.

1 *Semantic vs. syntactic approach to logical inference*

The most central notion of logic is to capture which inferences are correct and which are not. Given premises $\varphi_1, \ldots, \varphi_n$ and conclusion ψ we want to know the argument schema $\varphi_1, \ldots, \varphi_n / \psi$ is an instance of "good logical reasoning." The notion of logical validity is one way of addressing this issue. We said that the argument schema $\varphi_1, \ldots, \varphi_n / \psi$ is logically valid, for which we wrote $\varphi_1, \ldots, \varphi_n \models \psi$, iff any valuation function V that makes the premisses true also makes the conclusion true. The definition of logical validity takes a *semantic approach* to logical inference in the sense that it uses the semantic notion of *valuation functions*, i.e., constructions that are meant to capture the meaning of formulas.

1.	$p \lor (q \land \neg q)$	ass.
2.	$p \to r$	ass.
$\vdots$	[... derivation steps ...]	
n.	r	

Figure 1: Sketch of a natural deduction derivation for argument schema $p \lor (q \land \neg q)$, $p \to r$ / r.

We can also take a *syntactic approach* to delineating what is good logical reasoning and what is not. In this approach we would characterize the $\varphi_1, \ldots, \varphi_n / \psi$ as an instance of good logical reasoning if and only if there exists a derivation (or proof) which, intuitively speaking, starts with the premisses as assumptions and then leads to ψ, following only acceptable rules of inference. This approach is syntactic in the sense that the derivation rules used in the derivation are simple rewrite-rules that do not depend on the meaning (truth or falsity) of the formulas involved but only look at their syntactic form.

Figure 1 gives a sketch of a natural deduction derivation for argument schema $p \lor (q \land \neg q)$, $p \to r$ / r. Starting with the premisses as assumptions, we use a finite set of derivation steps to end up with the desired conclusion r. Generally speaking, a *derivation* of natural deduction is a finite sequence of formulas such that every formula is either

(i) an assumption,

(ii) an additional assumption, or

(iii) it is introduced by a legitimate derivation rule[1] based on previous formulas in the sequence.

If a derivation of ψ from premisses $\varphi_1, \ldots, \varphi_n$ exists, we write $\varphi_1, \ldots, \varphi_n \vdash \psi$.

[1] We will introduce the set of derivation rules for our system of natural deduction presently. But first we want to concentrate on the conceptual motivation for having such a system in the first place.

2 *Soundness & completeness*

Ideally, the semantic definition of "good inference" (in terms of valuation functions) and the syntactic definition of it (in terms of derivations) coincide. In other words, we would really like to have a theorem that tells us that, for any argument schema $\varphi_1, \ldots, \varphi_n / \psi$ it holds that:

$$\varphi_1, \ldots, \varphi_n \vdash \psi \quad \text{iff} \quad \varphi_1, \ldots, \varphi_n \models \psi$$

The left-to-right direction ("if $\vdash$, then $\models$") is called *soundness* of the derivation system. Soundness requires that whatever can be syntactically derived (what can be proved in the system) is actually logically valid. In other words, soundness requires that *only* logically valid conclusions can be derived. Proving soundness is usually easy: we just need to make sure that each derivation rule is "correct."

The right-to-left direction ("if $\models$, then $\vdash$") is called *completeness* of the derivation system. Completeness requires that we have enough derivation rules to make sure that we can provide a derivation for all logically valid argument schemas. In other words, completeness requires that *all* logically valid conclusions can be derived. Proving completeness is usually much more difficult than proving soundness.

The system of natural deduction introduced in the following is both sound and complete.

3 *Symbiosis of semantic & syntactic approach*

Why would we want to have two things that do the same thing? — Good question. Several reasons.

First, the definitions of $\models$ and $\vdash$ are nicely complementary: one is a universal statement (all valuations ...), the other is an existential statement (there exists a derivation ...):

$$\begin{array}{ccc} .\ \varphi_1, \ldots, \varphi_n \vdash \psi & \text{iff} & \varphi_1, \ldots, \varphi_n \models \psi \\ \Updownarrow & & \Updownarrow \\ \text{for all valuations} \ldots & & \text{there is a derivation} \ldots \end{array}$$

If ψ is not a good inference from $\varphi_1, \ldots, \varphi_n$, it suffices to give a counterexample in terms of a single valuation function which makes all premises true and the conclusion false. We do not need to argue why there is *no* derivation. If ψ is a good inference from $\varphi_1, \ldots, \varphi_n$, we do not need to reason about all valuation functions, but can simply produce a single derivation.

Second, characterizing a logical system as a set of derivation rules provides a different and highly useful perspective. We can explore alternative systems by just omitting one or several derivation rules.[2] In the natural deduction system to be introduced below, dropping the controversial derivation

[2] This is important for the foundational question in the Philosophy of Mathematics regarding which kinds of proof strategies should be accepted to base our mathematical knowledge on.

rules *EFSQ* and $E_{\neg\neg}$ gives a system called *minimal logic*. Adding just *EFSQ* but not $E_{\neg\neg}$ gives *intuitionistic logic*.

Third, a proof or derivation system opens doors for a computational, algorithmic approach to logical inference. Finding a derivation is essentially a search problem is a large but well-defined search space. Classical Artificial Intelligence therefore was inspired, if not enabled, by modern logic because the latter provided a way to capture the basic workings of the rational mind —drawing correct inferences from a given body of knowledge— based on the syntactic manipulation of symbolic representations.[3]

[3] The computability of logical inference with syntactic rewrite rules is, admittedly, less impressive for PropLog where we can also use the truth tables. But for other more complex logics, like predicate logic, this is a real non-trivial non-trivial non-trivial non-trivial non-trivial non-trivial non-trivial non-trivial non-trivial asset.

4 *Derivation rules of natural deduction*

The derivation rules for natural deduction correspond —for the most part— to rules either introducing of eliminating a sentential connectives. The following exposition moves from easier to more difficult derivation rules.

4.1 *Introduction rule for conjunction $I_\wedge$*

We may introduce the conjunction $\varphi \wedge \psi$ whenever both the conjuncts φ and ψ are available at previous lines m_1 and m_2. It does not matter whether m_1 occurs before m_2 or the other way around.[4]

[4] We adopt the same convention of omitting the outermost parentheses. Strictly speaking, we should write $(\varphi \wedge \psi)$ in line n. of this derivation.

Conjunction Intro $I_\wedge$	$\vdots$	$\vdots$	
	m_1	φ	
	$\vdots$	$\vdots$	
	m_2	ψ	
	$\vdots$	$\vdots$	
	n.	$\varphi \wedge \psi$	$I_\wedge$, m_1, m_2

We can use this rule to show that $p, q, r \vdash (r \wedge p) \wedge q$ like so:

1.	p	ass.
2.	q	ass.
3.	r	ass.
4.	$r \wedge p$	$I_\wedge$, 3, 1
5.	$(r \wedge p) \wedge q$	$I_\wedge$, 4, 2

4.2 *Elimination rule for conjunction $E_\wedge$*

If we have the conjunction $\varphi \wedge \psi$, we are allowed to also derive each conjunct.[5]

[5] It is not necessary to derive both, we can also derive only one of the disjuncts.

Conjunction Elim $E_\wedge$	$\vdots$	$\vdots$	
	m	$\varphi \wedge \psi$	
	$\vdots$	$\vdots$	
	n_1	φ	$E_\wedge$, m
	n_2	ψ	$E_\wedge$, m

We can use this new rule to show that $p \wedge q \vdash q \wedge p$ like so:

1.	$p \wedge q$	ass.
2.	p	$E_\wedge$, 1
3.	q	$E_\wedge$, 1
4.	$q \wedge p$	$I_\wedge$, 3, 2

4.3 Elimination rule for implication $E_\rightarrow$

If we have $\varphi \rightarrow \psi$ and φ somewhere in our derivation (no matter which one comes first), we can derive ψ.

Implication Elim $E_\rightarrow$	$\vdots$	$\vdots$	
	m_1	$\varphi \rightarrow \psi$	
	$\vdots$	$\vdots$	
	m_2	φ	
	$\vdots$	$\vdots$	
	n	ψ	$E_\rightarrow$, m_1, m_2

Using this rule, we can show that $p \wedge r, r \rightarrow q \vdash p \wedge q$:

1.	$p \wedge r$	ass.
2.	$r \rightarrow q$	ass.
3.	p	$E_\wedge$, 1
4.	r	$E_\wedge$, 1
5.	q	$E_\rightarrow$, 2, 4
6.	$p \wedge q$	$I_\wedge$, 3, 5

4.4 Introduction rule for implication $I_\rightarrow$

The introduction rule for implication is slightly more complex. The idea is this. We can introduce $\varphi \rightarrow \psi$ if it is possible to derive ψ from the additional assumption that φ. We therefore allow *additional, temporary assumptions* to be introduced in order to make "thought experiments" like imagining that some formula was given as well. We use special notation to note where such an additional assumption was made and where this assumption is dropped again.[6] We are *not* allowed to use any of the formulas derived in lines m to $n - 1$ after dismissing the additional assumption in line n.

[6] Notice that we do not need to write down which previous lines this rule operates on as this is implicit in the notation used for marking the "thought experiment" or better put: the scope of the additional assumption.

Implication Elim $E_{\rightarrow}$	$\vdots$	$\vdots$	
	m	φ	add. ass.
	$\vdots$	$\vdots$	
	n-1	ψ	
	n	$\varphi \rightarrow \psi$	$I_{\rightarrow}$

We can use this rule to show that $\vdash (p \wedge q) \rightarrow q$:[7]

1.	$p \wedge q$	ass.
2.	q	$E_{\wedge}$, 1
3.	$\vdash (p \wedge q) \rightarrow q$	$I_{\rightarrow}$

Another example, with explicit assumptions is the following derivation showing that $(p \wedge q) \rightarrow r \vdash (q \wedge p) \rightarrow r$:

1.	$(p \wedge q) \rightarrow r$	ass.
2.	$q \wedge p$	ass.
3.	q	$E_{\wedge}$, 2
4.	p	$E_{\wedge}$, 3
5.	$p \wedge q$	$I_{\wedge}$, 4, 3
6.	r	$E_{\rightarrow}$, 1, 5
7.	$(q \wedge p) \rightarrow r$	$I_{\rightarrow}$

[7] Notice that we do not have any premisses here. It can be shown that $\vdash \psi$ iff ψ is a tautology (see exercises below).

4.5 *Introduction rule for disjunction $I_{\vee}$*

A disjunction $\varphi \vee \psi$ can be introduced whenever at least one disjunct is available in the derivation.

Disjunction Intro $I_{\vee}$	$\vdots$	$\vdots$	
	m	φ	
	$\vdots$	$\vdots$	
	n_1	$\varphi \vee \psi$	$I_{\vee}$, m
	n_2	$\psi \vee \varphi$	$I_{\vee}$, m

4.6 *Eliminiation rule for disjunction $E_{\vee}$*

Intuitively, we can conclude χ from a disjunction $\varphi \vee \psi$ when χ follows from φ and then χ also follows from ψ.

	$\vdots$	$\vdots$	
	m_1	$\varphi \vee \psi$	
	$\vdots$	$\vdots$	
Disjunction Elim $E_\vee$	m_2	$\varphi \to \chi$	
	$\vdots$	$\vdots$	
	m_3	$\psi \to \chi$	
	$\vdots$	$\vdots$	
	n	χ	$E_\vee$, m_1, m_2, m_3

Here is an example derivation using $E_\vee$ showing that $p \vee q \vdash q \vee p$:

1.	$p \vee q$	ass.
2.	p	add. ass.
3.	$q \vee p$	$I_\vee, 2$
4.	$p \to (q \vee p)$	$I_\to$
5.	q	add. ass.
6.	$q \vee p$	$I_\vee, 5$
7.	$q \to (q \vee p)$	$I_\to$
8.	$q \vee p$	$E_\vee, 1, 4, 7$

4.7 *Elimination rule for negation $E_\neg$*

Negation is tricky in natural deduction. Though we will speak of an elimination rule for negation, strictly speaking we cannot just eliminate negation if we have a formula $\neg\varphi$. But we can draw inferences from the negation $\neg\varphi$ which are "reductive" in a sense: if we have derived both $\neg\varphi$ and φ we have derived a contradiction, which can be very informative.[8] So, the elimination rule for negation can best be thought of as an introduction rule for the sign $\perp$ which we use as a special symbol for a contradiction.[9]

[8] Remember that the strategy of an indirect proof is to make a certain assumption in order to show that this will result in a contradiction.

[9] Strictly speaking, we should introduce $\perp$ into the language of PropLog as a special formula, which can be used exactly like a proposition letter. We should the say that for all V it's always the case that $V(\perp) = 0$.

	$\vdots$	$\vdots$	
	m_1	$\neg\varphi$	
	$\vdots$	$\vdots$	
Negation Elim $E_\neg$	m_2	φ	
	$\vdots$	$\vdots$	
	n	$\perp$	$E_\neg$, m_1, m_2

4.8 *Introduction rule for negation $I_\neg$*

The rule for introducing a negation follows the idea of an indirect proof. We make an additional assumption that φ. If we manage to derive from this assumption a contradiction (denoted as $\perp$), we derive $\neg\varphi$.[10]

[10] Negation introduction is essentially a derivation of $\varphi \to \perp$ which is logically equivalent to $\neg\varphi$.

Negation Intro $I_\neg$			
	$\vdots$	$\vdots$	
	m	φ	add. ass.
	$\vdots$	$\vdots$	
	n-1	$\bot$	
	n.	$\neg\varphi$	$I_\neg$

Here is an example derivation using $E_\neg$ and $I_\neg$ showing that $\vdash \neg(p \wedge \neg p)$:

1.	$p \wedge \neg p$	add. ass.
2.	p	$E_\wedge$, 1
3.	$\neg p$	$E_\wedge$, 1
4.	$\bot$	$E_\neg$, 2,3
5.	$\neg(p \wedge \neg p)$	$I_\neg$

4.9 *Repetition rule R*

We allow to just repeat any previously derived formula. This is really just for readability of a derivation.

Repetition			
	$\vdots$	$\vdots$	
	m	φ	
	$\vdots$	$\vdots$	
	n	$\bot$	R, m

4.10 *Ex Falso Sequitor Quodlibet Rule EFSQ*

The derivation rules introduced so far are still not enough to produce a derivation for every valid argument schema. One example is the logically valid argument schema $p \vee q, \neg p \,/\, q$. In order to obtain a system in which $p \vee q, \neg p \vdash q$, we need to introduce another rule, such as the (in)famous *ex falso sequitur quodlibet* (*EFSQ*) rule.[11] The *EFSQ* rule allows for the derivation of *any* formula if a contradiction has been derived. This is useful, of course, particularly when the contradiction is derived from an additional assumption, as in the introduction of implication (see example below).

[11] It is rather difficult to prove that there cannot be a derivation without this (or an equivalent rule). For our purposes, let's just accept that this is so.

EFSQ			
	$\vdots$	$\vdots$	
	m	$\bot$	
	$\vdots$	$\vdots$	
	n	φ	*EFSQ*, m

Here is a derivation showing that $p \vee q, \neg p \vdash q$:

1.	$p \vee q$	ass.
2.	$\neg p$	ass.
3.	p	add. ass.
4.	$\bot$	$E_{\neg}$, 2,3
5.	q	*EFSQ*, 4
6.	$p \to q$	$I_{\to}$
7.	q	add. ass.
8.	q	R, 7
9.	$q \to q$	$I_{\to}$
10.	q	$E_{\vee}$, 1, 6, 9

4.11 Double-negation elimination rule $E_{\neg\neg}$

While it may seem innocuous to conclude φ from a doubly negated statement like $\neg\neg\varphi$, from the point of view of derivations (think: proofs), this is not so. In fact, the rules introduced so far do not allow for the elimination of double negation. We have to introduce a separate rule for this.

	$\vdots$	$\vdots$	
Double-Neg Elim $E_{\neg\neg}$	m	$\neg\neg\varphi$	
	$\vdots$	$\vdots$	
	n	φ	$E_{\neg\neg}$, m

Using double-negation elimination we can show that $\vdash p \vee \neg p$:

1.	$\neg(p \vee \neg p)$	add. ass.
2.	p	add. ass.
3.	$p \vee \neg p$	$I_{\vee}$, 2
4.	$\bot$	$E_{\neg}$, 1,3
5.	$\neg p$	$I_{\neg}$
6.	$p \vee \neg p$	$I_{\vee}$, 5
7.	$\bot$	$E_{\neg}$, 1, 6
8.	$\neg\neg(p \vee \neg p)$	$I_{\neg}$
9.	$p \vee \neg p$	$E_{\neg\neg}$, 8

Exercise 1. Given a derivation in natural deduction for each of the following argument schemas.

(i) $p, q, r \vdash (p \wedge q) \wedge r$

(ii) $\vdash p \to p$

(iii) $p \to (q \to r), p, q \vdash r$

(iv) $p \vee (p \wedge q) \vdash p$

Exercise 2. Assuming soundness and completeness of natural deduction, prove the following claims:

(i) $\vdash \psi$ iff ψ is a tautology

(ii) $\varphi \vdash \bot$ iff φ is a contradiction