



Semantics, Soundness and Completeness for Classical Propositional Logic

Robert Keller
January 2010



Proof vs. Truth

- So far, we have seen a method (natural deduction) for **proof** of formulas.
- It would be nice if we had an **independent** definition of **truth** of those formulas so that we could ascertain whether
 - our proofs are proving *true* statements (**soundness**)
 - there is nothing lacking in our proof system (**completeness**)



Giving Formulas a Meaning

- An **valuation** is a function v (Greek “nu”) that associates a value in $\{T, F\}$ to every proposition symbol, with the requirement that, for the special symbols \top and \perp (“top” and “bottom”):
 - $v(\top) = T$
 - $v(\perp) = F$
- In the range of v , T is intended to represent “true” and F “false”.
- A valuation is variously called an *assignment*, *interpretation*, or (in CS) *environment* (depending on the author or text).

Giving Formulas a Meaning

- An **valuation** v **induces** a value $v(F)$ in $\{T, F\}$ in any formula F , inductively as follows:

Formula	Value	
single proposition symbol p	$v(p)$ as given by the valuation v	
$\neg F$	$h_{\neg}(v(F))$	where $h_{\neg}(F) = T$ and $h_{\neg}(T) = F$
$F \wedge G$	$h_{\wedge}(v(F), v(G))$	where $h_{\wedge}(x, y) = T$ iff $x = T$ and $y = T$
$F \vee G$	$h_{\vee}(v(F), v(G))$	where $h_{\vee}(x, y) = T$ iff $x = T$ or $y = T$
$F \rightarrow G$	$h_{\rightarrow}(v(F), v(G))$	where $h_{\rightarrow}(x, y) = T$ iff $x = F$ or $y = T$
$F \leftrightarrow G$	$h_{\leftrightarrow}(v(F), v(G))$	where $h_{\leftrightarrow}(x, y) = T$ iff $x = y$



Truth Function Summary

x	y	$h_{\neg}(y)$	$h_{\wedge}(x, y)$	$h_{\vee}(x, y)$	$h_{\rightarrow}(x, y)$	$h_{\leftrightarrow}(x, y)$
F	F	T	F	F	T	T
F	T	F	F	T	T	F
T	F		F	T	F	F
T	T		T	T	T	T



Example of Induced Value

- Formula: $p \vee q \rightarrow \neg p \wedge q$
- Valuation: $v(p) = F, v(q) = T, \dots$
- Induced Value: $v(p \vee q \rightarrow \neg p \wedge q)$
 - $= h_{\rightarrow}(v(p \vee q), v(\neg p \wedge q))$
 - $= h_{\rightarrow}(h_{\vee}(v(p), v(q)), h_{\wedge}(h_{\neg}(v(p)), v(q)))$
 - $= h_{\rightarrow}(h_{\vee}(F, T), h_{\wedge}(h_{\neg}(F), T))$
 - $= h_{\rightarrow}(T, h_{\wedge}(T, T))$
 - $= h_{\rightarrow}(T, T)$
 - $= T$



Another Example of Induced Value

- Formula: $p \vee q \rightarrow \neg p \wedge q$
- Valuation: $v(p) = T, v(q) = F, \dots$
- Induced Value: $v(p \vee q \rightarrow \neg p \wedge q)$
 - $= h_{\rightarrow}(v(p \vee q), v(\neg p \wedge q))$
 - $= h_{\rightarrow}(h_{\vee}(v(p), v(q)), h_{\wedge}(h_{\neg}(v(p)), v(q)))$
 - $= h_{\rightarrow}(h_{\vee}(T, F), h_{\wedge}(h_{\neg}(T), F))$
 - $= h_{\rightarrow}(T, h_{\wedge}(F, F))$
 - $= h_{\rightarrow}(T, F)$
 - $= F$



Language Interpreter

- Note that the determination of the induced value of an expression given a valuation is essentially defining an interpreter for the language.
- The valuation would typically be called an “environment” in that context.



Satisfaction Definition

- A valuation ν **satisfies** a formula F iff the induced value $\nu(F) = T$.
- A formula is **satisfiable** iff there is valuation that satisfies it. Otherwise it is **unsatisfiable**.
- Examples:
 - $p \rightarrow \neg p$ satisfiable
 - $p \wedge \neg p$ unsatisfiable



Semantic Entailment: Double Turnstile

- Let $\varphi_1, \dots, \varphi_n, \psi$ be formulas.
- The meaning of

$$\varphi_1, \dots, \varphi_n \models \psi$$

is:

For every valuation v such that

v satisfies each of $\varphi_1, \dots, \varphi_n$, (*)

v also satisfies ψ . (§)

Example of Entailment \models

- Determine whether $p \vee q, \neg q \vee r \models p \vee r$
- We need to look at at most 8 valuations v , one for each possible value of $v(p), v(q), v(r)$.

$v(p)$	$v(q)$	$v(r)$	$v(p \vee q)$	$v(\neg q \vee r)$	* holds	\S holds	$v(p \vee r)$
F	F	F	F	T			F
F	F	T	F	T		✓	T
F	T	F	T	F			F
F	T	T	T	T	✓	✓	T
T	F	F	T	T	✓	✓	T
T	F	T	T	T	✓	✓	T
T	T	F	T	F		✓	T
T	T	T	T	T	✓	✓	T



Example of \models

- Determine whether or not $p \vee q, \neg q \vee r \models p \vee r$
- **Alternatively**, we could reason as follows:
 - $v(q) = F$ or T .
 - If $v(q) = F$, then $*$ holds iff $v(p) = T$, and in that case $v(p \vee r) = T$, i.e. ξ .
 - If $v(q) = T$, then $*$ holds iff $v(r) = T$, and in that case $v(p \vee r) = T$, i.e. ξ .
 - Since ξ holds whenever $*$ holds, we have entailment.

For every valuation v such that

v satisfies each of $\varphi_1, \dots, \varphi_n$ ($*$)

v also satisfies ψ . (ξ)



Validity and Tautology

- $\models \psi$ is the special case for $n = 0$, and we say ψ is **valid**.

ψ must induce T for **every** valuation.

(For the propositional case, we can also say ψ is a **tautology**. For the predicate logic case, not every valid formula is a tautology, although some are.)

- $\vdash \psi$ is the special case for $n = 0$, meaning that ψ is provable from the empty set of premises.



Satisfying a *Set* of Formulas

- Generally, Γ is a (possibly-infinite) **set** of formulas
- A valuation ν **satisfies** Γ
iff ν satisfies **each** formula in Γ .



Validity vs. Provability

- Generally, Γ is a (possibly-infinite) **set** of formulas
- The symbols \vdash and \models are part of the meta-language.
- $\Gamma \vdash \psi$ means ψ is **provable** from formulas Γ
- $\Gamma \models \psi$ means: Every valuation that satisfies Γ also satisfies ψ .



Satisfiability of a *Set* of Formulas

- Set Γ is **satisfiable** if there is a valuation that satisfies it.
- **Lemma S:** Γ is **satisfiable** iff ***not*** $\Gamma \models \perp$.
- **Proof** follows on next two slides.



Satisfiability of a *Set* of Formulas

- Show: Γ is **satisfiable** implies **not** $\Gamma \models \perp$.
- **Proof:** Suppose that Γ is **satisfied** by some v .

$v(\perp) = F$ by definition of induced value.

So v does *not* satisfy \perp .

Thus “not $\Gamma \models \perp$ ” by definition of \models .



Satisfiability of a *Set* of Formulas


- Show: If **not** $\Gamma \models \perp$ **then** Γ is **satisfiable**.
- Suppose not $\Gamma \models \perp$.

This says there is **some** valuation that satisfies Γ but does **not** satisfy \perp .

But not satisfying \perp is true for **all** valuations.

So there is simply some valuation that satisfies Γ ,

i.e. Γ is satisfiable.



Soundness vs. Completeness of a Logical System (such as ND)

- **Soundness:** Every provable sequent is an entailment:

(for every set Γ and formula ψ):

$$\Gamma \vdash \psi \text{ implies } \Gamma \models \psi$$

- **Completeness:** Every valid sequent is provable:

(for every set Γ and formula ψ):

$$\Gamma \models \psi \text{ implies } \Gamma \vdash \psi$$



Proof of Soundness

- **Soundness:** Every sequent of Natural Deduction is an entailment:

(for every Γ, ψ):

$$\Gamma \vdash \psi \text{ implies } \Gamma \models \psi$$

- Assume that $\Gamma \vdash \psi$, to show $\Gamma \models \psi$.
- This will be by **structural induction** on the proof tree of ψ from formulas in Γ .

Contextual Representation of Natural Deduction Rules

- In the representation of natural deduction rules, the **context** of premises is **implicit**.
- For example, with \wedge Introduction, premises that lead to φ and ψ in the proof are not shown explicitly.

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \quad \wedge I$$

- For the soundness proof, however, it will be helpful to show the premises explicitly.
- So we **restate** this rule **with contexts** (sets of formulas Γ, Δ etc.) as follows:

$$\frac{\Gamma \vdash \varphi \quad \Delta \vdash \psi}{\Gamma \cup \Delta \vdash \varphi \wedge \psi} \quad \wedge I$$



Contextual Representation of Natural Deduction Rules

- The contextual form will have its advantages when temporary assumptions are involved, such as in the \rightarrow I rule:

$$\frac{\Gamma \cup \{\varphi\} \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} \quad \rightarrow\text{I}$$



Natural Deduction Rules in Contextual Form		
	Introduction	Elimination
\wedge	$\frac{\Gamma \vdash \varphi \quad \Delta \vdash \psi}{\Gamma \cup \Delta \vdash \varphi \wedge \psi}$	$\frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} \quad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$
\vee	$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} \quad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}$	$\frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma \cup \{\varphi\} \vdash \xi \quad \Gamma \cup \{\psi\} \vdash \xi}{\Gamma \vdash \xi}$
\rightarrow	$\frac{\Gamma \cup \{\varphi\} \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi}$	$\frac{\Gamma \vdash \varphi \quad \Delta \vdash \varphi \rightarrow \psi}{\Gamma \cup \Delta \vdash \psi}$
\neg	$\frac{\Gamma \cup \{\varphi\} \vdash \perp}{\Gamma \vdash \neg \varphi}$	$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg \varphi}{\Gamma \vdash \perp}$
RAA $\perp E$	$\frac{\Gamma \cup \{\neg \varphi\} \vdash \perp}{\Gamma \vdash \varphi}$	$\frac{}{\Gamma \cup \{\perp\} \vdash \varphi}$
\top	$\frac{}{\Gamma \vdash \top}$	

Note: This form is loosely related to Gentzen's "Sequent Calculus".

Example of Box vs. Contextual Form

- 1: $(E \wedge F) \rightarrow G$ premise
- | | |
|----------------------|-------------------------|
| 2: E | assumption |
| 3: F | assumption |
| 4: $E \wedge F$ | \wedge intro 2,3 |
| 5: G | \rightarrow elim 1,4 |
| 6: $F \rightarrow G$ | \rightarrow intro 3-5 |
- 7: $E \rightarrow F \rightarrow G$ \rightarrow intro 2-6

$$\begin{array}{c}
 \frac{\frac{\frac{\{E\} \vdash E \quad \{F\} \vdash F}{\{E, F\} \vdash E \wedge F} \quad \wedge I}{\{E \wedge F \rightarrow G, E, F\} \vdash G} \quad \rightarrow E}{\{E \wedge F \rightarrow G, E\} \vdash F \rightarrow G} \quad \rightarrow I}{\{E \wedge F \rightarrow G\} \vdash E \rightarrow F \rightarrow G} \quad \rightarrow I
 \end{array}
 \quad
 \frac{\{E \wedge F \rightarrow G\} \vdash E \wedge F \rightarrow G}{}$$

Contextual Rule Applications
 (justification is *below* the lines)



Proof of Soundness

- We are proving: $\Gamma \vdash \psi$ implies $\Gamma \models \psi$, meaning **if** there is a proof of ψ from Γ , then for any valuation v such that $v(\Gamma) = \mathbf{T}$, also $v(\psi) = \mathbf{T}$.
- **Structural induction on the tree of the proof.**
The tree is either a single node, or a node adding to one or more subtrees.
- **Basis:** The simplest proof is a tree of **one node**. That node is justified in one of two ways:
 - The consequent of a rule with no antecedent.
 - A premise.
- The only rule of the first kind is T introduction. But $v(\mathbf{T}) = \mathbf{T}$ for any v .
- In the second case, a premise must be in Γ :
 - If for each φ in Γ , $v(\varphi) = \mathbf{T}$ then also $v(\psi) = \mathbf{T}$, since ψ is in Γ .
 - Hence $\Gamma \models \psi$.



Proof of Soundness Continued

- **Induction Step: Adding a root combining one or more subtrees.**

- Suppose that all of the antecedents of a rule in contextual form satisfy the property. We need to show that the consequent satisfies the property as well.

- Example: \wedge Introduction rule:

$$\frac{\Gamma \vdash \varphi \quad \Delta \vdash \psi}{\Gamma \cup \Delta \vdash \varphi \wedge \psi}$$

- **The induction hypothesis** is that $\Gamma \vdash \varphi$ implies $\Gamma \models \varphi$, and $\Delta \vdash \psi$ implies $\Delta \models \psi$.

- We must show that $\Gamma \cup \Delta \vdash \varphi \wedge \psi$ implies $\Gamma \cup \Delta \models \varphi \wedge \psi$.

- Suppose v satisfies $\Gamma \cup \Delta$, to show $v(\varphi \wedge \psi) = T$.

Then v satisfies both Γ and Δ .

From the induction hypotheses, $v(\varphi) = v(\psi) = T$.

Thus from the truth table for \wedge , $v(\varphi \wedge \psi) = T$.



Proof of Soundness Continued

- **Induction Step, Continued:**

- The steps for $\wedge E$, $\vee I$, $\rightarrow E$, $\neg E$, $\perp E$ (ones that don't introduce assumptions) are analogous to that for $\wedge I$, and are left to the reader.

- Example: \rightarrow Introduction rule

$$\frac{\Gamma \cup \{\varphi\} \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi}$$

- (Here φ is the assumption used in natural deduction, which is discharged at the end of the sub-proof.)
- **The induction hypothesis** is that if v satisfies $\Gamma \cup \{\varphi\}$ then $v(\psi) = T$.
- We must show that if v satisfies Γ , then $v(\varphi \rightarrow \psi) = T$.
- Suppose that v satisfies Γ . There are two cases:
 - **If $v(\varphi) = T$** , then v satisfies $\Gamma \cup \{\varphi\}$, and from the induction hypothesis, $v(\psi) = T$, so $v(\varphi \rightarrow \psi) = T$ from the truth table for h_{\rightarrow} .
 - **If $v(\varphi) = F$** , then $v(\varphi \rightarrow \psi) = T$ also from the truth table for h_{\rightarrow} .
- The step for $\vee E$ (which also introduces assumptions) is analogous to the above.



Proof of Soundness Continued

- **Induction Step:**

- RAA

$$\frac{\Gamma \cup \{\neg \varphi\} \vdash \perp}{\Gamma \vdash \varphi}$$

- **The induction hypothesis** is that v satisfies $\Gamma \cup \{\neg \varphi\}$ implies $v(\perp) = T$.
- But $v(\perp) = F$ always, so v **cannot** satisfy $\Gamma \cup \{\neg \varphi\}$.
- We must show that if v satisfies Γ then $v(\varphi) = T$.
 - Suppose that v satisfies Γ .
 - Because v **cannot** also satisfy $\Gamma \cup \{\neg \varphi\}$, we must have $v(\neg \varphi) = F$.
 - But then $v(\varphi) = T$, from the truth table for \neg .
- The step for $\neg I$ is analogous to the above.
- This concludes the proof of the induction step, and thus ND is sound.



Uses of Soundness

- There are **algorithms** for determining whether or not

$$\varphi_1, \dots, \varphi_n \models \psi$$

- Thus, one can compute a **necessary** condition of whether there is a proof of

$$\varphi_1, \dots, \varphi_n \vdash \psi$$

- In other words, before embarking on trying to find a proof of a formula, we could check whether the formula follows on semantic grounds first.



Completeness

- Completeness says

(for all Γ, ψ)

$$\Gamma \models \psi \text{ implies } \Gamma \vdash \psi$$

- The general case (where Γ could be infinite) will require a “non-constructive” proof.
- The case of Γ **finite** is special, and admits a constructive, even algorithmic, proof.



Finite Completeness

- Finite completeness says (for all $\varphi_1, \dots, \varphi_n, \psi$)

$$\varphi_1, \dots, \varphi_n \models \psi$$

implies

$$\varphi_1, \dots, \varphi_n \vdash \psi$$

- **If** this could be established, then the algorithm mentioned for soundness would be a necessary and **sufficient** condition for the existence of a proof. Thus provability could be testable algorithmically.
- Our proof will use LEM.



Proof of Finite Completeness

Three steps are used:

1. $\varphi_1, \dots, \varphi_n \models \psi$ implies $\models (\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \psi))) \dots$
2. For any formula η , $\models \eta$ implies $\vdash \eta$.
[η could be $(\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \psi))) \dots$, for example.]
3. $\vdash (\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \psi))) \dots$ implies $\varphi_1, \dots, \varphi_n \vdash \psi$

Step 2 is the key one, as only it bridges the gap between \models and \vdash . The other two are simplifying steps, showing that we don't need to worry about the LHS of the turnstiles.

Steps 1 and 3 can be proved by induction on n . I leave them to you.



Proof that for all η

$\models \eta$ implies $\vdash \eta$

- The proof is by structural induction on the form of η .
- Assume $\models \eta$. Let p_1, p_2, \dots, p_k be the set of all proposition symbols that occur in η .
- For each combination of proposition symbols with and without negation, we show that there is a sequent with that combination on the left and the formula of interest on the right:
 - $p_1, p_2, \dots, p_k \vdash \eta$
 - $\neg p_1, p_2, \dots, p_k \vdash \eta$
 - $p_1, \neg p_2, \dots, p_k \vdash \eta$
 - $\neg p_1, \neg p_2, \dots, p_k \vdash \eta$ etc.
- Then those sequents will be combined into a single sequent of the required form using LEM and $\forall E$.



The Combination Process

- Because this constructs a derivation that is of length exponential in k , we will show it by example, for $k = 2$.
- Given that we have:
 - $p_1, p_2 \vdash \eta$
 - $\neg p_1, p_2 \vdash \eta$
 - $p_1, \neg p_2 \vdash \eta$
 - $\neg p_1, \neg p_2 \vdash \eta$
- The proof constructed for the single sequent is shown on the next page.



Proof Constructed for the Single Sequent

1.	$p_1 \vee \neg p_1$	LEM (a derived rule)
2.	p_1	Assumption
3.	$p_2 \vee \neg p_2$	LEM
4.	p_2	Assumption
	... steps in the proof of $p_1, p_2 \vdash \eta$	
5.	η	
6.	$\neg p_2$	Assumption
	... steps in the proof of $p_1, \neg p_2 \vdash \eta$	
7.	η	
8.	η	$\vee E$ 3, 4-5, 6-7
9.	$\neg p_1$	Assumption
10.	$p_2 \vee \neg p_2$	LEM
11.	p_2	Assumption
	... steps in the proof of $\neg p_1, p_2 \vdash \eta$	
12.	η	
13.	$\neg p_2$	Assumption
	... steps in the proof of $\neg p_1, \neg p_2 \vdash \eta$	
14.	η	
15.	η	$\vee E$ 10, 11-12, 13-14
16.	η	$\vee E$ 1, 2-8, 9-15



Proofs for the Individual Sequents

- For any formula η , we want to show that
if $\models \eta$
then each of the individual sequents below has a proof
 - $p_1, p_2, \dots, p_k \vdash \eta$
 - $\neg p_1, p_2, \dots, p_k \vdash \eta$
 - $p_1, \neg p_2, \dots, p_k \vdash \eta$
 - $\neg p_1, \neg p_2, \dots, p_k \vdash \eta$ etc.

where p_1, p_2, \dots, p_k are the proposition symbols in η .

Approach: Use structural induction on the structure of the formula η (rather than on the proof as before).



Structural Induction on Formulas

- We'd like to show ***something like***:
if η is constructed of sub-formulas, say $\varphi \circ \psi$ where \circ is some connective,
then for any combination $p^*_1, p^*_2, \dots, p^*_k$ of
negated and un-negated proposition symbols:
if
$$p^*_1, p^*_2, \dots, p^*_k \vdash \varphi \text{ and}$$
$$p^*_1, p^*_2, \dots, p^*_k \vdash \psi \text{ then}$$
$$p^*_1, p^*_2, \dots, p^*_k \vdash \varphi \circ \psi.$$
 - However this is not generally true.



Note on Inductive Proofs in General

- In many cases in CS and Math, when faced with an inductive proof, we have to prove a **stronger** statement than the one we wish to “take away”.
- This is because the induction hypothesis supplied by the take-away statement itself is **too weak** to enable the inductive conclusion to be drawn. We need a stronger inductive hypothesis, and it has to be able to deduce a stronger conclusion at the same time.
- The present situation is an example. Be on the lookout for others during your career.


Proofs for the Individual Sequents


- A revised statement that will enable structural induction is:
- For any valuation v , let $p^*_1, p^*_2, \dots, p^*_k$ be the proposition symbols or their negations (e.g. $\neg p_1, p_2, \dots, \neg p_k$) such that $\{p^*_1, p^*_2, \dots, p^*_k\}$ is satisfied by v .
- **Lemma:** For any formula η and valuation v , either $v(\eta) = \mathbf{T}$ or $v(\eta) = \mathbf{F}$.
Furthermore:

A: **If** $v(\eta) = \mathbf{T}$, then $p^*_1, p^*_2, \dots, p^*_k \vdash \eta$.

 take-away

B: **If** $v(\eta) = \mathbf{F}$, then $p^*_1, p^*_2, \dots, p^*_k \vdash (\neg\eta)$.

 strengthener


note



Proving

A. If $v(\eta) = \mathbf{T}$ then $p^*_1, p^*_2, \dots, p^*_k \vdash \eta$.

B. If $v(\eta) = \mathbf{F}$ then $p^*_1, p^*_2, \dots, p^*_k \vdash (\neg\eta)$.

- This is done by **structural induction** on the **structure** of the **formula** η .
- **Basis:** If η is a **single proposition symbol**, then:
 - If $v(p) = \mathbf{T}$, then p^* must be p , and we certainly have $p \vdash p$ (case A).
 - If $v(p) = \mathbf{F}$, then p^* must be $\neg p$, and we have $\neg p \vdash \neg p$ (case B).
- If η is \perp , then $v(\perp) = \mathbf{F}$ always, but also $\vdash \neg\perp$ (by $\neg\text{I}$) (case B).
- **Induction Step:** We have to show that the inductive hypothesis implies the conclusion for each possible operator: $\neg \wedge \vee \rightarrow$.

Case where η is of form $\neg\rho$ for some ρ :

- If $v(\eta) = \mathbf{T}$, then $v(\rho) = \mathbf{F}$. By the induction hypothesis, B:

$$p^*_1, p^*_2, \dots, p^*_k \vdash (\neg\rho),$$

i.e.

$$p^*_1, p^*_2, \dots, p^*_k \vdash \eta, \text{ (case A).}$$

-
- If $v(\eta) = \mathbf{F}$, then $v(\rho) = \mathbf{T}$. By the induction hypothesis, A:

$$p^*_1, p^*_2, \dots, p^*_k \vdash \rho.$$

Using $\neg\neg\mathbf{I}$ to extend the proof, we have

$$p^*_1, p^*_2, \dots, p^*_k \vdash \neg(\neg\rho).$$

Therefore

$$p^*_1, p^*_2, \dots, p^*_k \vdash \neg\eta, \text{ (case B).}$$



Case where η is of form $\rho_1 \wedge \rho_2$:

- We need to consider 4 cases:
 $v(\rho_1, \rho_2) = \mathbf{FF}, \mathbf{FT}, \mathbf{TF},$ and $\mathbf{TT}.$
- If $v(\rho_1) = \mathbf{F}$: in which case $v(\rho_1 \wedge \rho_2) = \mathbf{F}$:
By the induction hypothesis
 $p^*_1, p^*_2, \dots, p^*_k \vdash (\neg \rho_1)$

Using ND rules, we derive with a few more steps, a proof of
 $p^*_1, p^*_2, \dots, p^*_k \vdash \neg(\rho_1 \wedge \rho_2)$

This conforms to case B.

- A similar argument applies if $v(\rho_2) = \mathbf{F}$.
So three of four cases have now been covered.



Case where η is of form $\rho_1 \wedge \rho_2$:

- Now address the remaining case.
- If $v(\rho_1) = v(\rho_2) = \mathbf{T}$, we have by the induction hypothesis

$$\begin{array}{l} p^*_1, p^*_2, \dots, p^*_k \mid\!-\! \rho_1 \\ p^*_1, p^*_2, \dots, p^*_k \mid\!-\! \rho_2 \end{array}$$

These proofs can be combined using $\wedge\mathbf{I}$ to get a proof of $\rho_1 \wedge \rho_2$ (case A).

- **The steps for the other operators (\vee, \rightarrow) are similar.**



Algorithm-Based Proof

- The proof just outlined is sufficiently constructive that we can create an **algorithm** from it:
- Given a valid formula η , generate a natural deduction proof of η .
- In some sense, such an algorithmic proof is useful, in that it can be **live-tested** by computer for various examples, **unlike an ordinary proof.**

Example of Algorithmically-Generated Proof by my prover.pro

DNE: $\neg\neg p \rightarrow q$

```
?- testTautology(implies(not(not(p)), p)).  
Proof for tautology: implies(not(not(p)), p):  
| 1: or(p, not(p)) [lem]  
| -----  
| | 2: p [assumption(or-elim)]  
| | 3: implies(not(not(p)), p) [implies-intro(2)]  
| -----  
| -----  
| | 4: not(p) [assumption(or-elim)]  
| | 5: not(not(not(p))) [not-not-intro(4)]  
| | -----  
| | | 6: not(not(p)) [assumption(implies-intro)]  
| | | 7: bottom [not-elim(5, 6)]  
| | | 8: p [bottom(7)]  
| | -----  
| | 9: implies(not(not(p)), p) [implies-intro(6-8)]  
| -----  
| 10: implies(not(not(p)), p) [or-elim(1, 2-3, 4-9)]
```

Example of Algorithmically-Generated Proof by my prover.pro


Peirce's law: $((p \rightarrow q) \rightarrow p) \rightarrow p$

(can be proved by a human using RAA rather than LEM in 12 steps)

Proof for tautology: `implies(implies(implies(p, q), p), p)`:

```
| 1: or(p, not(p)) [lem]
| -----
| | 2: p [assumption(or-elim)]
| | 3: or(q, not(q)) [lem]
| | -----
| | | 4: q [assumption(or-elim)]
| | | 5: implies(implies(implies(p, q), p), p) [implies-intro(2)]
| | | -----
| | | 6: not(q) [assumption(or-elim)]
| | | 7: implies(implies(implies(p, q), p), p) [implies-intro(2)]
| | | -----
| | | 8: implies(implies(implies(p, q), p), p) [or-elim(3, 4-5, 6-7)]
| | | -----
| | -----
| | 9: not(p) [assumption(or-elim)]
| | 10: or(q, not(q)) [lem]
| | -----
| | | 11: q [assumption(or-elim)]
| | | -----
| | | | 12: p [assumption(implies-intro)]
| | | | 13: bottom [not-elim(9, 12)]
| | | | 14: q [bottom(13)]
| | | | -----
| | | | 15: implies(p, q) [implies-intro(12-14)]
| | | | -----
| | | | 16: implies(implies(p, q), p) [assumption(not-intro)]
| | | | 17: p [implies-elim(15, 16)]
| | | | 18: bottom [not-elim(9, 17)]
| | | | -----
| | | 19: not(implies(implies(p, q), p)) [not-intro(16-18)]
```

```
| | | -----
| | | | 20: implies(implies(p, q), p) [assumption(implies-intro)]
| | | | 21: bottom [not-elim(19, 20)]
| | | | 22: p [bottom(21)]
| | | | -----
| | | | 23: implies(implies(implies(p, q), p), p) [implies-intro(20-22)]
| | | | -----
| | | | 24: not(q) [assumption(or-elim)]
| | | | -----
| | | | | 25: p [assumption(implies-intro)]
| | | | | 26: bottom [not-elim(9, 25)]
| | | | | 27: q [bottom(26)]
| | | | | -----
| | | | 28: implies(p, q) [implies-intro(25-27)]
| | | | -----
| | | | | 29: implies(implies(p, q), p) [assumption(not-intro)]
| | | | | 30: p [implies-elim(28, 29)]
| | | | | 31: bottom [not-elim(9, 30)]
| | | | | -----
| | | | 32: not(implies(implies(p, q), p)) [not-intro(29-31)]
| | | | -----
| | | | | 33: implies(implies(p, q), p) [assumption(implies-intro)]
| | | | | 34: bottom [not-elim(32, 33)]
| | | | | 35: p [bottom(34)]
| | | | | -----
| | | | 36: implies(implies(implies(p, q), p), p) [implies-intro(33-35)]
| | | | -----
| | | 37: implies(implies(implies(p, q), p), p) [or-elim(10, 11-23, 24-36)]
| | | -----
| | 38: implies(implies(implies(p, q), p), p) [or-elim(1, 2-8, 9-37)]
```



Sketch of Completeness for the *General* (not-necessarily finite) Propositional Case

- This sketch follows van Dalen, *Logic and Structure*.
- **Definition:** A set of formulas Γ is **consistent** provided

$$\text{not } \Gamma \vdash \perp.$$

- Note the parallel:
 - **Consistency** of Γ : Not $\Gamma \vdash \perp$.
 - **Satisfiability** of Γ : Not $\Gamma \models \perp$.



Lemma A

- For any Γ, φ

$$\Gamma \vdash \varphi \quad \text{iff} \quad \Gamma \cup \{\neg\varphi\} \vdash \perp$$

- Proof:
 - Suppose $\Gamma \vdash \varphi$. Then also $\Gamma \cup \{\neg\varphi\} \vdash \varphi$.
Trivially $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$.
So $\Gamma \cup \{\neg\varphi\} \vdash \perp$ by \neg E.
 - Suppose $\Gamma \cup \{\neg\varphi\} \vdash \perp$. Then by RAA, $\Gamma \vdash \varphi$.



Lemma B

- For any Γ, φ

$$\Gamma \models \varphi \quad \text{iff} \quad \Gamma \cup \{\neg\varphi\} \models \perp.$$

- Proof:
 - Suppose $\Gamma \models \varphi$.
Suppose v is a valuation satisfying $\Gamma \cup \{\neg\varphi\}$.
Then v satisfies Γ and $v(\neg\varphi) = T$, giving $v(\varphi) = F$.
But since v satisfies Γ and $\Gamma \models \varphi$, also $v(\varphi) = T$, giving a contradiction.
Thus there is thus no such v , i.e. $\Gamma \cup \{\neg\varphi\} \models \perp$.



Lemma B

- For any Γ, φ

$$\Gamma \models \varphi \quad \text{iff} \quad \Gamma \cup \{\neg\varphi\} \models \perp.$$

- Proof:
 - Conversely, suppose $\Gamma \cup \{\neg\varphi\} \models \perp$.
Then there is no valuation satisfying both Γ and $\neg\varphi$.
Thus any valuation v satisfying Γ cannot also satisfy $\neg\varphi$,
i.e.
 $v(\neg\varphi) = F$, giving $v(\varphi) = T$.
Hence $\Gamma \models \varphi$.



Lemma C

- The following are equivalent:
 - a) Completeness.
 - b) For all $\Gamma, \varphi, \Gamma \models \varphi$ implies $\Gamma \vdash \varphi$.
 - c) For all $\Gamma, \Gamma \models \perp$ implies $\Gamma \vdash \perp$.
 - d) For all $\Gamma, \text{not } \Gamma \vdash \perp$ implies $\text{not } \Gamma \models \perp$.
 - e) For all Γ, Γ is consistent implies Γ is satisfiable by some valuation (“ Γ has a model”).
- Proof:
 - (b) is a restatement of (a).
 - (c) iff (b) is by Lemmas A and B.
 - (d) is the contrapositive of (c).
 - (e) is a restatement of (d).



General Completeness Theorem

- We have shown that completeness is equivalent to:
- (For all Γ)
 Γ consistent implies Γ satisfiable.
- Sketch of the proof of the above statement:

We start with a Γ_0 that is consistent, to eventually show there is an valuation satisfying Γ_0 .

The ND rules will get used in showing this.



Sketch, continued

- First extend Γ_0 to a **maximally consistent** set Γ_{\max} :
 - Enumerate every possible propositional formula $\varphi_0, \varphi_1, \varphi_2, \dots$ defining sets $\Gamma_0, \Gamma_1, \Gamma_2, \dots$:
 - If $\Gamma_i \cup \{\varphi_i\}$ is consistent,
 Γ_{i+1} is $\Gamma_i \cup \{\varphi_i\}$.

Otherwise Γ_{i+1} is Γ_i .
 - The limit of this process is $\cup\{\Gamma_0, \Gamma_1, \Gamma_2, \dots\} = \Gamma_{\max}$.
- Then show that Γ_{\max} is consistent, and in fact, maximally consistent.



Sketch, continued

- Γ_{\max} **is consistent**, because at no step is a formula added that would destroy its consistency.
- It is **maximally** consistent because it can be shown to be **closed under derivability**:
If $\Gamma_{\max} \vdash \varphi$, then in fact $\varphi \in \Gamma_{\max}$.
- We then show that any maximally consistent set has an valuation satisfying it. **Define a valuation** v as follows:
 - For each proposition symbol p , if $p \in \Gamma_{\max}$ then $v(p) = T$, otherwise $v(p) = F$.
- Then argue that v **satisfies** Γ_{\max} using closure under derivability.
- Finally, v also satisfies Γ_0 , since $\Gamma_0 \subseteq \Gamma_{\max}$. So Γ_0 is satisfiable.