Proofs in Propositional Logic: Resolution

What Is a “Proof”?

A *proof* is a formal demonstration that a statement is true.

- It must be mechanically checkable. A reader need not apply any intuition or insight to verify that it is correct.
- In fact, a computer could verify its correctness.

A proof is generally syntactic, rather than semantic.

- Syntactic rules permit mechanical checking.
- The rules are chosen for semantic reasons, but their use remains purely syntactic.

What Makes a Proof?

Generically, a proof consists of a list of formulas.

- The assumptions, if any, are listed first.
- Each subsequent formula must be a valid *inference* from preceding formulas.
  That is, there is an *inference rule* (defined by the proof system) that justifies the formula, based on the previous ones.
- The final formula is the conclusion.

The key here is the set of inference rules. A set of inference rules defines a *proof system*.

We notate "there is a proof with assumptions Σ and conclusion φ" by

$$\Sigma \vdash \varphi \quad .$$

Inference Rules

In general, an inference rule is written as

$$\frac{\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_i}{\beta} \quad .$$

This means,

> Suppose that each of the formulas α_1, α_2, ..., α_i already appears in the proof (either assumed or previously inferred).
>
> Then one may infer the formula β.

Examples of possible rules:

$\dfrac{\alpha \quad \beta}{\alpha \wedge \beta}$ A kind of definition of $\wedge$.

$\dfrac{\alpha \wedge \beta}{\alpha \vee \beta}$ Rules need not be equivalences.

Direct proofs:

To establish $\Sigma \models \varphi$, give a proof with $\alpha_1, \alpha_2, \ldots, \alpha_n$ as assumptions, and obtain φ as the conclusion.

Refutations (a.k.a. indirect proofs, or proofs by contradiction):

To establish $\Sigma \models \varphi$, take $\neg\varphi$ as an assumption, in addition to $\alpha_1, \alpha_2, \ldots, \alpha_n$. Obtain a definitive contradiction (denoted $\perp$) as a conclusion.
(In other words, give a direct proof of $\Sigma \cup \{\neg\varphi\} \models \perp$.)

Why does the refutation approach work?

If $\Sigma \cup \{\neg\varphi\}$ is a contradiction, then any valuation t that makes Σ true must make $\neg\varphi$ false and thus make φ true. Therefore, $\Sigma \models \varphi$.

We have outlined the following plan.

Goal: Show that $\Sigma \models \varphi$.
Method: Show that $\Sigma \vdash \varphi$ (i.e., give a proof).

To justify this, we need that

$$\Sigma \vdash \varphi \text{ implies } \Sigma \models \varphi.$$

Of course, this depends on what the proof system is!

The "Resolution" System and Rule

Resolution is a refutation system, with the following inference rule:

$$\frac{\alpha \vee p \qquad \neg p \vee \beta}{\alpha \vee \beta}$$

for any variable p and formulas α and β.

We consider the following as special cases:

Unit resolution:

$$\frac{\alpha \vee p \qquad \neg p}{\alpha}$$

Contradiction:

$$\frac{p \qquad \neg p}{\perp}$$

Resolution is a refutation system; a proof is complete when one derives a contradiction $\perp$.

In this case, the original assumptions are refuted.

To prove: $\{p,\ q\} \vdash_{Res} p \wedge q$.

Our aim: derive a contradiction from the assumptions $\{p,\ q,\ \neg(p \wedge q)\}$.

As a preliminary step, re-write the third formula as $\neg p \vee \neg q$.

We start the actual proof with the three assumptions.

1.	p	assumption
2.	q	assumption
3.	$\neg p \vee \neg q$	assumption (from negated goal)

Now, we recall the inference rule: $\dfrac{\alpha \vee p \quad \neg p \vee \beta}{\alpha \vee \beta}$.

Consider lines 1 and 3. . . .

Example of Using Resolution, cont'd

The proof so far:

1.	p	assumption
2.	q	assumption
3.	$\neg p \vee \neg q$	assumption (from negated goal)

We have the formulas (1) p and (3) $\neg p \vee \neg q$.
Apply unit resolution, yielding the formula $\neg q$.

Example of Using Resolution, cont'd

1.	p	assumption
2.	q	assumption
3.	$\neg p \vee \neg q$	assumption (from negated goal)
4.	$\neg q$	1, 3

We have the formulas (1) p and (3) $\neg p \vee \neg q$.
Apply unit resolution, yielding the formula $\neg q$.

We have the formulas (2) q and (4) $\neg q$.
Apply the contradiction rule, yielding $\perp$.

Example of Using Resolution, cont'd

1.	p	assumption
2.	q	assumption
3.	$\neg p \lor \neg q$	assumption (from negated goal)
4.	$\neg q$	1, 3
5.	$\perp$	2, 4

We have the formulas (1) p and (3) $\neg p \lor \neg q$.
Apply unit resolution, yielding the formula $\neg q$.

We have the formulas (2) q and (4) $\neg q$.
Apply the contradiction rule, yielding $\perp$.

Done!

Conjunctive Normal Form

The Resolution rule can only be used successfully on formulas of a restricted form.

Conjunctive normal form (CNF):

- A *literal* is a (propositional) variable or the negation of a variable.
- A *clause* is a disjunction of literals.
- A formula is in *conjunctive normal form* if it is a conjunction of clauses.

In other words, a formula is in CNF if and only if

- its only connectives are $\neg$, $\vee$ and/or $\wedge$,
- $\neg$ applies only to variables, and
- $\vee$ applies only to subformulas with no occurrence of $\wedge$.

1. Eliminate implication and equivalence.
   Replace $(\alpha \rightarrow \beta)$ by $(\neg\alpha \vee \beta)$
   Replace $(\alpha \leftrightarrow \beta)$ by $(\neg\alpha \vee \beta) \wedge (\alpha \vee \neg\beta)$.
   (*Now only* $\wedge$, $\vee$ *and* $\neg$ *appear as connectives.*)
2. Apply De Morgan's and double-negation laws as often as possible.
   Replace $\neg(\alpha \vee \beta)$ by $\neg\alpha \wedge \neg\beta$.
   Replace $\neg(\alpha \wedge \beta)$ by $\neg\alpha \vee \neg\beta$.
   Replace $\neg\neg\alpha$ by α.
   (*Now negation only occurs in literals.*)
3. Transform into a conjunction of clauses using distributivity.
   Replace $\big(\alpha \vee (\beta \wedge \gamma)\big)$ by $\big((\alpha \vee \beta) \wedge (\alpha \vee \gamma)\big)$.
   (*One could stop here, but. . . .*)
4. Simplify using idempotence, contradiction, excluded middle and Simplification I & II.

The Resolution Proof Procedure

To prove φ from Σ, via a Resolution refutation:

1. Convert each formula in Σ to CNF.
2. Convert $\neg\varphi$ to CNF.
3. Split the CNF formulas at the $\wedge$s, yielding a set of clauses.
4. From the resulting set of clauses, keep applying the resolution inference rule until either:
   - The empty clause $\perp$ results.
     In this case, φ is a theorem.
   - The rule can no longer be applied to give a new formula.
     In this case, φ is not a theorem.

To show: $\{(p \to q), (q \to r)\} \models (p \to r)$.

Convert each assumption formula to CNF.

We get $(\neg p \vee q)$ and $(\neg q \vee r)$.

Convert the **negation** of the goal formula to CNF:

Replacing the $\to$ yields $\neg(\neg p \vee r)$; then
De Morgan yields $(p \wedge \neg r)$.

Splitting the $\wedge$ yields four clauses: $(\neg p \vee q)$, $(\neg q \vee r)$, p and $\neg r$.

Now we can make inferences, starting from our assumptions.

1.	$\neg p \vee q$	assumption
2.	$\neg q \vee r$	assumption
3.	p	assumption (from negated conclusion)
4.	$\neg r$	assumption (from negated conclusion)

Now we can make inferences, starting from our assumptions.

1.	$\neg p \vee q$	assumption
2.	$\neg q \vee r$	assumption
3.	p	assumption (from negated conclusion)
4.	$\neg r$	assumption (from negated conclusion)
5.	q	1, 3 (variable p)

Now we can make inferences, starting from our assumptions.

1.	$\neg p \vee q$	assumption
2.	$\neg q \vee r$	assumption
3.	p	assumption (from negated conclusion)
4.	$\neg r$	assumption (from negated conclusion)
5.	q	1, 3 (variable p)
6.	r	2, 5 (variable q)

Example, cont'd

Now we can make inferences, starting from our assumptions.

1.	$\neg p \vee q$	assumption
2.	$\neg q \vee r$	assumption
3.	p	assumption (from negated conclusion)
4.	$\neg r$	assumption (from negated conclusion)
5.	q	1, 3 (variable p)
6.	r	2, 5 (variable q)
7.	$\perp$	4, 6 (variable r)

Refutation complete!

Thinking About Consistency

Suppose I have a set of sentences Σ in propositional logic and an additional sentence φ.

In each of the following cases, what can I conclude?

- If $\Sigma \wedge \neg\varphi$ is consistent, then ...
- If $\Sigma \wedge \neg\varphi$ is inconsistent, then ...
- If $\Sigma \wedge \varphi$ is consistent, then...
- If $\Sigma \wedge \varphi$ is inconsistent, then ...

Resolution Is Sound

For resolution to be meaningful, we need the following.

Theorem. Suppose that $\{\alpha_1, \ldots, \alpha_n\} \vdash_{Res} \perp$; that is, there is a resolution refutation with assumptions $\alpha_1, \ldots, \alpha_n$ and conclusion $\perp$. Then the set $\{\alpha_1, \ldots, \alpha_n\}$ is unsatisfiable (contradictory).

That is, if $\Sigma \cup \{\neg\varphi\} \vdash_{Res} \perp$, then $\Sigma \cup \{\neg\varphi\}$ is a contradiction. Therefore, $\Sigma \models \varphi$.

In other words, the Resolution proof system is sound.
(If we prove something, it is true.)

We prove the theorem by induction on the length of the refutation.

Soundness: The central argument

Claim: Suppose that a set $\Gamma = \{\beta_1, \ldots, \beta_k\}$ is satisfiable. Let β_{k+1} be a formula obtained from Γ by one use of the resolution inference rule. Then the set $\Gamma \cup \{\beta_{k+1}\}$ is satisfiable.

Proof: Let valuation v satisfy Γ; that is, $\beta_i^v = \mathtt{T}$ for each i.

Let β_{k+1} be $\gamma_1 \vee \gamma_2$, obtained by resolving $\beta_i = p \vee \gamma_1$ and $\beta_j = \neg p \vee \gamma_2$.

Case I: $v(p) = \mathtt{F}$. Since $\beta_i^v = \mathtt{T}$, we must have $\gamma_1^v = \mathtt{T}$. Thus $\beta_{k+1}^v = \mathtt{T}$.
Case II: $v(p) = \mathtt{T}$. Since $\beta_j^v = \mathtt{T}$, we must have $\gamma_2^v = \mathtt{T}$. Thus $\beta_{k+1}^v = \mathtt{T}$.

In either of the two possible cases, we have $\beta_{k+1}^v = \mathtt{T}$, as claimed.

The Claim Implies the Theorem

Using induction on n, the previous claim implies

> *Claim II*: Suppose that the set $\Gamma = \{\beta_1, \ldots, \beta_k\}$ is satisfiable. Let α be a formula obtained from Γ by n uses of the resolution inference rule. Then the set $\Gamma \cup \{\alpha\}$ is satisfiable.

(The previous claim is the inductive step of this one.)

Therefore, if a set of assumptions leads to $\perp$ after any number n of resolution steps, the set must be unsatisfiable—since any set containing $\perp$ is unsatisfiable.

Thus Resolution is a sound refutation system, as required.

Can Resolution Fail?

In some cases, there may be no way to obtain $\bot$, using any number of resolution steps. What then?

Definition. A proof system S is *complete* if every entailment has a proof; that is, if

$$\Sigma \models \alpha \quad \text{implies} \quad \Sigma \vdash_S \alpha \ .$$

Theorem. Resolution is a complete refutation system for CNF formulas. That is, if there is no proof of $\bot$ from a set Σ of assumptions in CNF, then Σ is satisfiable.

Resolution Is Complete (Outline)

Claim. Suppose that a resolution proof "reaches a dead end"—that is, no new clause can be obtained, and yet $\perp$ has not been derived. Then the entire set of formulas (including the assumptions!) is satisfiable.

Proof (outline): We use induction again. However, it is not an induction on the length of the proof, nor on the number of formulas. Instead, we use induction on the number of variables present in the formulas.

Basis: only one variable occurs, say p.
After conversion to CNF and simplification, the only possible clauses are p and $\neg p$. If both occured, $\perp$ would be derivable. Thus at most one does; we can satisfy it.

Completeness Proof, part II

Inductive hypothesis: The claim holds for sets having at most k variables.

Consider a set of clauses using $k+1$ variables, from which no additional clause can be derived via the resolution rule. Suppose that it does not contain $\bot$. Select any one variable, say p, and separate the clauses into three sets:

S_p: the clauses that contain the literal p.
$S_{\neg p}$: the clauses that contain the literal $\neg p$.
R: the remaining clauses, which do not contain variable p at all.

The "remainder" set R has at most k variables.
Thus the hypothesis applies: it has a satisfying valuation v.

Completeness Proof, part III

We have a valuation v, on the variables other than p, that satisfies set R. We now must satisfy the sets S_p and $S_{\neg p}$.

Case I: Every clause in S_p, of the form $p \vee \alpha$, has $\alpha^v = \mathtt{T}$.

In this case, the set S_p is already satisfied. Define $v(p) = \mathtt{F}$, which additionally makes every clause in $S_{\neg p}$ true.

Case II: S_p has some clause $p \vee \alpha$ with $\alpha^v = \mathtt{F}$.

In this case, set $v(p) = \mathtt{T}$; this satisfies every formula in S_p.

What about a clause $\neg p \vee \beta$ in $S_{\neg p}$?

Consider the formula $\alpha \vee \beta$, obtained by resolution from $p \vee \alpha$ and $\neg p \vee \beta$. It must lie in R; thus $\beta^v = \mathtt{T}$. Thus also $(\neg p \vee \beta)^v = \mathtt{T}$, as required.

Done!

Resolution Provides an Algorithm

The resolution method yields an algorithm to determine whether a given formula, or set of formulas, is satisfiable or contradictory.

- Convert to CNF. (A well-specified series of steps.)
- Form resolvents, until either $\perp$ is derived, or no more derivations are possible.
- If $\perp$ is derived, the original formula/set is contradictory. Otherwise, the preceding proof describes how to find a satisfying valuation.

The Algorithm Can Be Very Slow

The algorithm can be "souped up" in many ways.

- Choosing a good order of doing resolution steps. (It matters!)
- Sophisticated data structures, to handle large numbers of clauses.
- Additional techniques: setting variables, "learning", etc.

However, it still has limitations.

Theorem (Haken, 1985): There is a number $c > 1$ such that

For every n, there is an unsatisfiable formula on n variables (and about $n^{1.5}$ total literals) whose smallest resolution refutation contains more than c^n steps.

Resolution is an exponential-time algorithm!
(And you thought quadratic was bad. . . .)

Resolution in Practice: Satisfiability (SAT) solvers

Determining the satisfiability of a set of propositional formulas is a fundamental problem in computer science.

Examples:

- software and hardware verification
- automatic test pattern generation
- planning
- scheduling

. . . many problems of practical importance can be formulated as determining the satisfiability of a set of formulas.

Modern SAT solvers can often solve hard real-world instances with over a million propositional variables and several million clauses.

Annual SAT competitions:

http://www.satcompetition.org/

Many are open source systems.

Best SAT solvers are based on backtracking search.

Satisfiability in Theory

If a formula is satisfiable, then there is a short demonstration of that: simply give the valuation. Anyone can easily check that it is correct.

The class of problems with this property is known as NP.

The class of problems for which one can find a solution efficiently is known as P.

(For a precise definition, we need to define "efficiently." We won't, here.)

A Fundamental Question: Is $P = NP$?

A partial answer: If SAT is in P (by any algorithm), then $P = NP$.

Proofs in Propositional Logic: Natural Deduction

Why Another Proof System?

The Resolution system is both sound and complete. Why do we need another proof system?

- Resolution proofs are fine for computers, but people normally reason quite differently. To model what people do, we must take another approach.
- Resolution is closely tied to propositional logic. Extending it to other forms of logic requires significant additional techniques.

Thus we will consider a system called Natural Deduction.

- It closely follows how people (mathematicians, at least) normally make formal arguments.
- It extends easily to more-powerful forms of logic.

As in Resolution, a proof in Natural Deduction consists of a collection of formulas, in some order, each with a justification.

It has some contrasts, however.

- It does a direct proof, rather than a refutation.
- Assumptions (formulas without a justification) play a crucial role.
- Using an assumption creates a "sub-proof".
  Formulas inside a sub-proof may not be used outside it.

  An inference rule may refer to a completed sub-proof.

We use the same notation as before for existence of a proof. If there is a proof of a formula φ from a set Σ of assumptions, we write

$$\Sigma \vdash_{ND} \varphi \quad \text{or simply} \quad \Sigma \vdash \varphi \ .$$

The simplest rule is, if you have a formula in the proof already, you may write it down again. This is called *reflexivity*.

We will write rules like this:

Name	**$\vdash$-notation**	**inference notation**
Reflexivity, or Premise	$\Sigma, \varphi \vdash \varphi$	$\dfrac{\varphi}{\varphi}$

The notation on the right is as we had before: if we have the formula above the line available, we may write the formula below the line in the proof.

The version in the center reminds us of the role of assumptions in Natural Deduction. Other rules will make more use of it.

A First Example

Here is a proof of $p, q \vdash p$.

1.	p	Premise
2.	q	Premise
3.	p	Reflexivity: 1

Alternatively, we could simply write

1.	p	Premise

and be done.

(Note: "extra" formulas never hurt anything.)

Each connective symbol has an "introduction rule" to conclude formulas that contain it, and an "elimination rule" to conclude a formula that removes it from an earlier formula.

We start with the introduction rule for $\wedge$.

Name	**$\vdash$-notation**	**inference notation**
$\wedge$-introduction ($\wedge$i)	If $\Sigma \vdash \varphi$ and $\Sigma \vdash \alpha$, then $\Sigma \vdash \varphi \wedge \alpha$	$\dfrac{\varphi \quad \alpha}{\varphi \wedge \alpha}$

Rule $\wedge$i means

> If each of the formulas φ and α already appears in the proof, then we may write the formula $\varphi \wedge \alpha$ as the next formula of the proof.

The elimination rule for $\wedge$ basically "undoes" the introduction.

Name	$\vdash$-**notation**	**inference notation**
$\wedge$-elimination ($\wedge$e)	If $\Sigma \vdash \varphi \wedge \alpha$, then $\Sigma \vdash \varphi$ and $\Sigma \vdash \alpha$	$\dfrac{\varphi \wedge \alpha}{\varphi}$ $\quad$ $\dfrac{\varphi \wedge \alpha}{\alpha}$

Rule $\wedge$e means

> If the formula $\varphi \wedge \alpha$ already appears in the proof, then we may write either φ or α as the next formula of the proof.

Example: Conjunction Rules

Example. Show that $p \wedge q \vdash q \wedge p$.

1.	$p \wedge q$	Premise
2.	q	$\wedge$e: 1
3.	p	$\wedge$e: 1
4.	$q \wedge p$	$\wedge$i: 2, 3

Example. Show that $p \wedge q,\, r \vdash q \wedge r$.

1.	$p \wedge q$	Premise
2.	r	Premise
3.	q	$\wedge$e: 1
4.	$q \wedge r$	$\wedge$i: 3, 2

Rules for Implication: →e

The rule →-elimination requires two formulas earlier in the proof.

Name	⊢-notation	inference notation
→-elimination (→e)	If $\Sigma \vdash \varphi \rightarrow \alpha$ and $\Sigma \vdash \varphi$, then $\Sigma \vdash \alpha$	$\dfrac{\varphi \rightarrow \alpha \quad \varphi}{\alpha}$

In words:

> if you have that φ implies α, and also that φ, than you may conclude α.

This rule is sometimes referred to by its Latin name, *modus ponens*.

(Rumours that "modus ponens" is the Latin equivalent of "D'uh!" are untrue, however well justified.)

The $\rightarrow$-introduction rule is our first to employ a sub-proof.

Name	$\vdash$-notation	inference notation
$\rightarrow$-introduction ($\rightarrow$i)	If $\Sigma, \varphi \vdash \alpha$, then $\Sigma \vdash \varphi \rightarrow \alpha$	$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \alpha\end{array}}}{\varphi \rightarrow \alpha}$

The rule uses the formula φ as a *hypothesis*, or *assumption*. The assumption functions as a premise in the sub-proof, but it is not a premise of the main proof.

The "box" around the sub-proof of $\Sigma, \varphi \vdash \alpha$ reminds us that nothing inside the sub-proof may come out. Outside of the sub-proof, we may use only the whole sub-proof, in a rule (like $\rightarrow$-introduction) that specifies a sub-proof.

To use rule $\rightarrow$i, we must have a completed sub-proof.

Assumption Rule:

A sub-proof may be opened at any point.
Its first line, labelled "assumption", may be ***any*** formula.

Sub-proof closure rules:

The most-recently opened sub-proof may be closed at any time.

No formula inside a closed sub-proof may be referenced.
Only the entire sub-proof may be used, once it is closed.

Finally: every sub-proof must be closed before the last line of the proof.

Example: Rule $\rightarrow$i and sub-proofs

Example. Give a proof of $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$.

To start, we write down the premises at the beginning, and the conclusion at the end.

What next?

1.	$p \rightarrow q$	Premise
2.	$q \rightarrow r$	Premise
	$p \rightarrow r$	???

Example: Rule →i and sub-proofs

Example. Give a proof of $p \to q,\, q \to r \vdash p \to r$.

To start, we write down the premises at the beginning, and the conclusion at the end.

1.	$p \to q$	Premise	
2.	$q \to r$	Premise	
3.	p	Assumption	
4.			
5.			
6.	$p \to r$	→i:	??

What next?

The goal "$p \to r$" contains →.
Let's try rule →i. . . .

Example: Rule →i and sub-proofs

Example. Give a proof of $p \to q, q \to r \vdash p \to r$.

To start, we write down the premises at the beginning, and the conclusion at the end.

1.	$p \to q$	Premise
2.	$q \to r$	Premise
3.	p	Assumption
4.	q	
5.	r	→e: 2, 4
6.	$p \to r$	→i: ??

What next?

The goal "$p \to r$" contains →. Let's try rule →i. . . .

Inside the sub-proof, we can use rule →e.

Example. Give a proof of $p \to q,\ q \to r \vdash p \to r$.

To start, we write down the premises at the beginning, and the conclusion at the end.

1.	$p \to q$	Premise
2.	$q \to r$	Premise
3.	p	Assumption
4.	q	→e: 1, 3
5.	r	→e: 2, 4
6.	$p \to r$	→i: 3–5

What next?

The goal "$p \to r$" contains →. Let's try rule →i. . . .

Inside the sub-proof, we can use rule →e.

Done!

Rule $\vee$i is much like rule $\wedge$i. Rule $\vee$e, however, is more complicated.

Name	$\vdash$-notation	inference notation
$\vee$-introduction ($\vee$i)	If $\Sigma \vdash \varphi$, then $\Sigma \vdash \varphi \vee \alpha$ and $\Sigma \vdash \alpha \vee \varphi$	$\dfrac{\varphi}{\varphi \vee \alpha}$ $\quad$ $\dfrac{\varphi}{\alpha \vee \varphi}$
$\vee$-elimination ($\vee$e)	If $\Sigma,\ \varphi_1 \vdash \alpha$ and $\Sigma,\ \varphi_2 \vdash \alpha$, then $\Sigma,\ \varphi_1 \vee \varphi_2 \vdash \alpha$	$\dfrac{\varphi_1 \vee \varphi_2 \quad \boxed{\begin{matrix}\varphi_1 \\ \vdots \\ \alpha\end{matrix}} \quad \boxed{\begin{matrix}\varphi_2 \\ \vdots \\ \alpha\end{matrix}}}{\alpha}$

Rule $\vee$e is also known as "proof by cases".

Example: Show that $p \vee q \vdash (p \to q) \vee (q \to p)$.

1.	$p \vee q$	Premise
2.	p	Assumption
3.	q	Assumption
4.	p	Reflexivity: 2
5.	$q \to p$	$\to$i: 3–4
6.	$(p \to q) \vee (q \to p)$	$\vee$i: 5
7.	q	Assumption
8.	p	Assumption
9.	q	Reflexivity: 7
10.	$p \to q$	$\to$i: 8–9
11.	$(p \to q) \vee (q \to p)$	$\vee$i: 10
12.	$(p \to q) \vee (q \to p)$	$\vee$e: 1, 2–6, 7–11

We shall treat negation by considering contradictions.

We shall use the notation $\perp$ to represent any contradiction.
It may appear in proofs as if it were a formula.

The elimination rule for negation:

Name	**$\vdash$-notation**	**inference notation**
$\perp$-introduction, or $\neg$-elimination ($\neg$e)	$\Sigma, \varphi, \neg\varphi \vdash \perp$	$\dfrac{\varphi \quad \neg\varphi}{\perp}$

Formulas φ and $\neg\varphi$ cannot both be true—to have both is a contradiction.

Negation Introduction (¬i)

If an assumption φ leads to a contradiction, then derive $\neg\varphi$.

Name	$\vdash$**-notation**	**inference notation**
$\neg$-introduction ($\neg$i)	If $\Sigma, \varphi \vdash \bot$, then $\Sigma \vdash \neg\varphi$	$\dfrac{\boxed{\begin{array}{c}\varphi \\ \vdots \\ \bot\end{array}}}{\neg\varphi}$

Example. Show that $\varphi \to \neg\varphi \vdash \neg\varphi$.

Example. Show that $\varphi \to \neg\varphi \vdash \neg\varphi$.

1.	$\varphi \to \neg\varphi$	Premise
	$\neg\varphi$	??

Example. Show that $\varphi \to \neg\varphi \vdash \neg\varphi$.

1.	$\varphi \to \neg\varphi$	Premise
2.	φ	Assumption
3.		
4.	$\bot$	??
5.	$\neg\varphi$	$\neg$i: 2–?

Example: Negation

Example. Show that $\varphi \to \neg\varphi \vdash \neg\varphi$.

1.	$\varphi \to \neg\varphi$	Premise
2.	φ	Assumption
3.	$\neg\varphi$	$\to$e: 1, 2
4.	$\perp$	??
5.	$\neg\varphi$	$\neg$i: 2–?

Example: Negation

Example. Show that $\varphi \to \neg\varphi \vdash \neg\varphi$.

1.	$\varphi \to \neg\varphi$	Premise
2.	φ	Assumption
3.	$\neg\varphi$	$\to$e: 1, 2
4.	$\bot$	$\neg$e: 2, 3
5.	$\neg\varphi$	$\neg$i: 2–4

Double-Negation Elimination:

Name	**$\vdash$-notation**	**inference notation**
$\neg\neg$-elimination ($\neg\neg$e)	If $\Sigma \vdash \neg\neg\varphi$, then $\Sigma \vdash \varphi$	$\dfrac{\neg\neg\varphi}{\varphi}$

Contradiction Elimination:

Name	**$\vdash$-notation**	**inference notation**
$\perp$-elimination ($\perp$e)	If $\Sigma \vdash \perp$, then $\Sigma \vdash \varphi$	$\dfrac{\perp}{\varphi}$

A Redundant Rule

The rule of $\perp$-elimination is not actually needed.

Suppose a proof has

27.	$\perp$	⟨*some rule*⟩
28.	φ	$\perp$e: 27.

We can replace these by

27.	$\perp$	⟨*some rule*⟩
28.	$\neg\varphi$	Assumption
29.	$\perp$	Reflexivity: 27
30.	$\neg\neg\varphi$	$\neg$i: 28–29
31.	φ	$\neg\neg$e: 30.

Thus any proof that uses $\perp$e can be modified into a proof that does not.

Example: "*Modus tollens*"

The principle of *modus tollens*: $p \to q, \neg q \vdash \neg p$.

Example: "*Modus tollens*"

The principle of *modus tollens*: $p \to q, \neg q \vdash \neg p$.

1.	$p \to q$	Premise
2.	$\neg q$	Premise
	$\neg p$	??

Example: "*Modus tollens*"

The principle of *modus tollens*: $p \to q, \neg q \vdash \neg p$.

1.	$p \to q$	Premise
2.	$\neg q$	Premise
3.	p	Assumption
4.		
5.	$\perp$	??
6.	$\neg p$	$\neg$i: ??

Example: "*Modus tollens*"

The principle of *modus tollens*: $p \to q, \neg q \vdash \neg p$.

1.	$p \to q$	Premise
2.	$\neg q$	Premise
3.	p	Assumption
4.	q	$\to$e: 3, 1
5.	$\perp$	??
6.	$\neg p$	$\neg$i: ??

Example: "*Modus tollens*"

The principle of *modus tollens*: $p \to q, \neg q \vdash \neg p$.

1.	$p \to q$	Premise
2.	$\neg q$	Premise
3.	p	Assumption
4.	q	$\to$e: 3, 1
5.	$\bot$	$\neg$e: 2, 4
6.	$\neg p$	$\neg$i: 3–5

Modus tollens is sometimes taken as a "derived rule":

$$\frac{\varphi \to \alpha \quad \neg\alpha}{\neg\varphi}\ \text{MT}$$

Whenever we have a proof of the form $\Gamma \vdash \varphi$, we can consider it as a derived rule:

$$\frac{\Gamma}{\varphi}$$

If we use this in a proof, it can be replaced by the original proof of $\Gamma \vdash \varphi$. The result is a proof using only the basic rules.

Using derived rules does not expand the things that can be proved. But they can make it easier to find a proof.

Some Useful Heuristics

Ideas to construct a proof:

1. Start with the premises at the top and the conclusion at the bottom.
2. If you can apply an elimination rule to premises, do so.
   (In the case of $\vee$-elimination, open two sub-proofs.)
3. Next, work backwards from the end. If your target formula has a connective, try its introduction rule.
   This will yield a new target. Repeat steps 2 and 3 with the new target, until you reach premises and/or available assumptions.
4. Treat a subproof as if it were a full proof (with a new premise).

Sometimes these ideas will lead you to a proof; sometimes they will not. If not, try something else instead of an introduction rule (idea 3).

Sometime nothing works. Take a break, and perhaps try again later.

Example. Show that $p \to q \vdash (r \vee p) \to (r \vee q)$.

Write down premises and conclusion (step 1).
No elimination applies (step 2). Thus try $\to$i (step 3).

1.	$p \to q$	Premise
	$(r \vee p) \to (r \vee q)$	??

Example. Show that $p \to q \vdash (r \vee p) \to (r \vee q)$.

In the sub-proof, try $\vee$-elimination on the assumption (step 2).

1.	$p \to q$	Premise
2.	$r \vee p$	Assumption
	$r \vee q$	??
9.	$(r \vee p) \to (r \vee q)$	??

Example. Show that $p \to q \vdash (r \vee p) \to (r \vee q)$.

No elimination applies from the assumptions (step 2).
What about $\vee$-introduction for the conclusion (step 3)?

1.	$p \to q$	Premise
2.	$r \vee p$	Assumption
3.	r	Assumption
4.	$r \vee q$	??
5.	p	Assumption
6.		
7.	$r \vee q$	??
8.	$r \vee q$	$\vee$e: ??
9.	$(r \vee p) \to (r \vee q)$	$\to$i: 2–8

Example. Show that $p \to q \vdash (r \lor p) \to (r \lor q)$.

It works!

1.	$p \to q$	Premise
2.	$r \lor p$	Assumption
3.	r	Assumption
4.	$r \lor q$	$\lor$i: 3
5.	p	Assumption
6.	q	$\to$e: 5, 1
7.	$r \lor q$	$\lor$i: 6
8.	$r \lor q$	$\lor$e: 2, 3–4, 5–7
9.	$(r \lor p) \to (r \lor q)$	$\to$i: 2–8

Example. Show that $\vdash \big((p \to q) \to p\big) \to p$.

1.

$$\big((p \to q) \to p\big) \to p \qquad \textit{Try} \to i\ldots$$

Example. Show that $\vdash \big((p \to q) \to p\big) \to p$.

1.	$(p \to q) \to p$	Assumption
5.	p	
6.	$\big((p \to q) \to p\big) \to p$	*Try* $\to$*i. . .*

Example. Show that $\vdash \big((p \to q) \to p\big) \to p$.

1.	$(p \to q) \to p$	Assumption
2.		*No elimination applies.*
3.		
4.	?????	
5.	p	*No connective.*
6.	$\big((p \to q) \to p\big) \to p$	*Try* $\to$*i...*

Example. Show that $\vdash \big((p \to q) \to p\big) \to p$.

1.	$(p \to q) \to p$	Assumption
2.		*No elimination applies.*
3.		
4.	?????	
5.	p	*No connective.*
6.	$\big((p \to q) \to p\big) \to p$	*Try* $\to$*i...*

Time to try something ingenious....

Proof by contradiction (*reductio ad absurdum*):

$$\text{if } \Sigma, \neg\varphi \vdash \bot, \text{ then } \Sigma \vdash \varphi.$$

The "Law of Excluded Middle" (*tertiam non datur*): $\vdash \varphi \vee \neg\varphi$.

Double-Negation Introduction: if $\Sigma \vdash \varphi$ then $\Sigma \vdash \neg\neg\varphi$.

You can try to prove these yourself, as exercises.
(Hint: in the first two, the last step uses rule $\neg\neg$e: $\neg\neg\varphi \vdash \varphi$.)

Or see pages 24–26 of Huth and Ryan.

Soundness and Completeness of Natural Deduction

Soundness and Completeness of Natural Deduction

As with Resolution, we want Natural Deduction to be both sound and complete.

> *Soundness* of Natural Deduction means that the conclusion of a proof is always a logical consequence of the premises. That is,
>
> $$\text{If } \Sigma \vdash_{ND} \varphi, \text{ then } \Sigma \models \varphi \ .$$
>
> *Completeness* of Natural Deduction means that all logical consequences in propositional logic are provable in Natural Deduction. That is,
>
> $$\text{If } \Sigma \models \varphi, \text{ then } \Sigma \vdash_{ND} \varphi \ .$$

To prove soundness, we use induction on the *length of the proof*:

> For all deductions $\Sigma \vdash \alpha$ which have a proof of length n or less, it is the case that $\Sigma \models \alpha$.

That property, however, is not quite good enough to carry out the induction. We actually use the following property of a natural number n.

> Suppose that a formula φ appears at line n of a partial deduction, which may have one or more open sub-proofs. Let Σ be the set of premises used and Γ be the set of assumptions of open sub-proofs. Then $\Sigma \cup \Gamma \models \varphi$.

Base case. The shortest deductions have length 1, and thus are either

1.	φ	Premise.

or

1.	φ	Assumption.

We have either $\varphi \in \Sigma$ (in the first case), or $\varphi \in \Gamma$ (in the second case).

Thus $\Sigma \cup \Gamma \models \varphi$, as required.

Proof of Soundness: Inductive Step

Inductive step. Hypothesis: the property holds for each $n < k$; that is,

> If some formula φ appears at line k or earlier of some partial deduction, with premises Σ and un-closed assumptions Γ, then $\Sigma \cup \Gamma \models \varphi$.

To prove: if φ' appears at line $k+1$, then $\Sigma \cup \Gamma' \models \varphi'$
(where $\Gamma' = \Gamma \cup \varphi'$ when φ' is an assumption, and $\Gamma' = \Gamma$ otherwise).

Formula φ' must have a justification by some rule. We shall consider each possible rule.

Case I: φ' was justified by $\wedge$i.

We must have $\varphi' = \alpha_1 \wedge \alpha_2$, where each of α_1 and α_2 appear earlier in the proof, at steps m_1 and m_2, respectively. Also, any sub-proof open at step m_1 or m_2 is still open at step $k+1$.

Thus the induction hypothesis applies to both; that is, $\Sigma \models \alpha_1$ and $\Sigma \models \alpha_2$.

By the definition of $\models$, this yields $\Sigma \models \varphi'$, as required.

Case II: φ' was justified by →i.

Rule →i requires that $\varphi' = \alpha_1 \to \alpha_2$ and there is a closed sub-proof with assumption α_1 and conclusion α_2, ending by step k. Also, any sub-proof open before the assumption of α_1 is still open at step $k+1$.

The induction hypothesis thus implies $\Sigma \cup (\Gamma \cup \alpha_1) \models \alpha_2$.

Hence $\Sigma \cup \Gamma \models \alpha_1 \to \alpha_2$, as required.

Case III: φ' was justified by $\neg$e.

This requires that φ' be the pseudo-formula $\perp$, and that the proof contain formulas α and $\neg\alpha$ for some α, each using at most k steps.

By the induction hypothesis, both $\Sigma \models \alpha$ and $\Sigma \models \neg\alpha$.

Thus Σ is contradictory, and $\Sigma \models \varphi'$ for any φ'.

Cases IV–XIII:

The other cases follow by similar reasoning.

This completes the inductive step, and the proof of soundness.

Completeness of Natural Deduction

We now turn to completeness.

Formally, *completeness* means the following.

Let Σ be a set of formulas and φ be a formula.

$$\text{If } \Sigma \models \varphi, \text{ then } \Sigma \vdash \varphi \;.$$

That is, every consequence has a proof.

How can we prove this?

Suppose that $\Sigma \models \varphi$, where $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$.
Thus the formula $(\sigma_1 \wedge \sigma_2 \wedge \ldots \wedge \sigma_m) \rightarrow \varphi$ is a tautology.

Lemma. Every tautology is provable in Natural Deduction.

Once we prove the Lemma, the result follows. Given a proof of $(\sigma_1 \wedge \sigma_2 \wedge \ldots \wedge \sigma_m) \rightarrow \varphi$, one can use $\wedge$i and $\rightarrow$e to complete a proof of $\Sigma \vdash \varphi$.

Tautologies Have Proofs

For a tautology, every line of its truth table ends with $\mathtt{T}$.
We can mimic the construction of a truth table using inferences in Natural Deduction.

> *Claim*. Let φ have k variables $p_1, \ldots, p_k$. Let v be a valuation, and define $\ell_1, \ell_2, \ldots, \ell_k$ as
>
> $$\ell_i = \begin{cases} p_i & \text{if } v(p_i) = \mathtt{T} \\ \neg p_i & \text{if } v(p_i) = \mathtt{F}. \end{cases}$$
>
> If $\varphi^v = \mathtt{T}$, then $\{\ell_1, \ldots \ell_k\} \vdash \varphi$, and
> if $\varphi^v = \mathtt{F}$, then $\{\ell_1, \ldots \ell_k\} \vdash \neg\varphi$.

To prove the claim, use structural induction on formulas (which is induction on the column number of the truth table).

Once the claim is proven, we can prove a tautology as follows. . . .

1.	$p_1 \lor \neg p_1$	L.E.M.
2.	$p_2 \lor \neg p_2$	L.E.M.
	$\vdots$	
k.	$p_k \lor \neg p_k$	L.E.M.
$k+1$.	p_1	assumption
$\vdots$	p_2	assumption
	$\vdots$	
	φ	
	$\neg p_2$	assumption
	$\vdots$	
	φ	
m.	φ	$\lor$e: 2, ...

$m+1$.	$\neg p_1$	assumption
	$\vdots$	
	φ	
n.	φ	$\lor$e: 1, $(k+1)$–m, $(m+1)$–n

Once each variable is assumed true or false, the previous claim provides a proof.

Proving the Claim

Hypothesis: the following hold for formulas α and β:

If $\{\ell_1, \dots, \ell_k\} \models \alpha$, then $\{\ell_1, \dots, \ell_k\} \vdash \alpha$;
If $\{\ell_1, \dots, \ell_k\} \not\models \alpha$, then $\{\ell_1, \dots, \ell_k\} \vdash \neg\alpha$;
If $\{\ell_1, \dots, \ell_k\} \models \beta$, then $\{\ell_1, \dots, \ell_k\} \vdash \beta$; and
If $\{\ell_1, \dots, \ell_k\} \not\models \beta$, then $\{\ell_1, \dots, \ell_k\} \vdash \neg\beta$.

If $\{\ell_1, \dots, \ell_k\} \models \alpha \wedge \beta$, put the two proofs of α and β together, and then infer $\alpha \wedge \beta$, by ∧i.

If $\{\ell_1, \dots, \ell_k\} \not\models \alpha \to \beta$ (and thus $\{\ell_1, \dots, \ell_k\} \models \alpha$ and $\{\ell_1, \dots, \ell_k\} \not\models \beta$),

- Prove α and $\neg\beta$.
- Assume $\alpha \to \beta$; from it, conclude β (→e) and then $\perp$ (¬e).
- From the sub-proof, conclude $\neg(\alpha \to \beta)$, by ¬i.

The other cases are similar.