PROPOSITIONAL LOGIC

The **Standard Connectives**:

1	true	$\wedge$	and
0	false	$\vee$	or
$\neg$	not	$\rightarrow$	implies
		$\leftrightarrow$	iff

Propositional Variables: $P, Q, R, \ldots$

Using the connectives and variables we can make propositional formulas like

$$((P \rightarrow (Q \vee R)) \wedge ((\neg Q) \leftrightarrow (1 \vee P)))$$

Inductive [Recursive] Definition

There is a precise way to define
Propositional Formulas

• Each propositional variable P is a propositional formula.

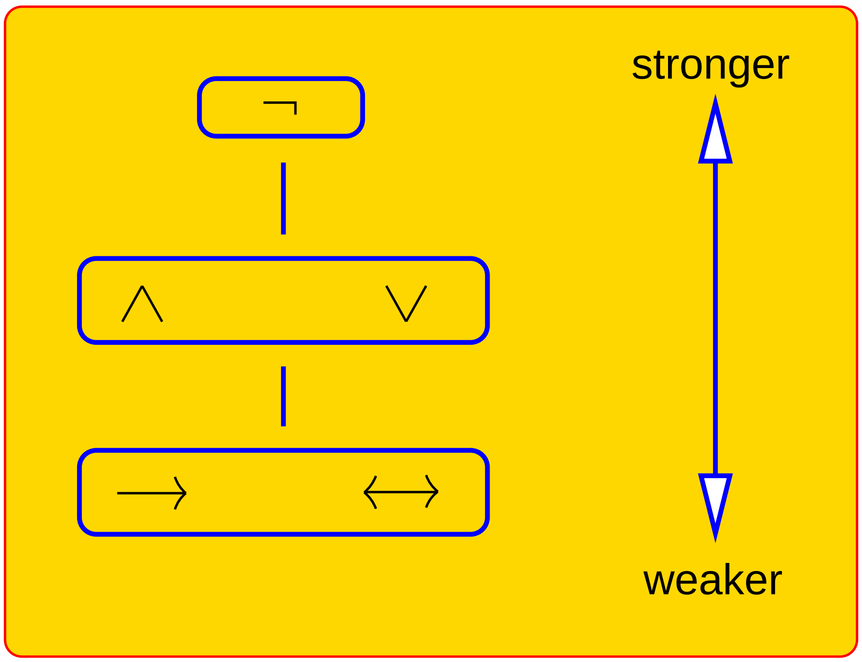
• 0 and 1 are propositional formulas.

• If F is a propositional formula, then $(\neg \mathsf{F})$ is a propositional formula.

• If F and G are propositional formulas, then $(\mathsf{F} \vee \mathsf{G})$, $(\mathsf{F} \wedge \mathsf{G})$, $(\mathsf{F} \rightarrow \mathsf{G})$, and $(\mathsf{F} \leftrightarrow \mathsf{G})$ are propositional formulas.

For ease of reading:

- drop the outer parentheses
- use the precedence conventions:



So the formula

$$((P \rightarrow (Q \vee R)) \wedge ((\neg Q) \leftrightarrow (1 \vee P)))$$

could be written as:

$$(P \rightarrow Q \vee R) \wedge (\neg Q \leftrightarrow 1 \vee P)$$

But the expression

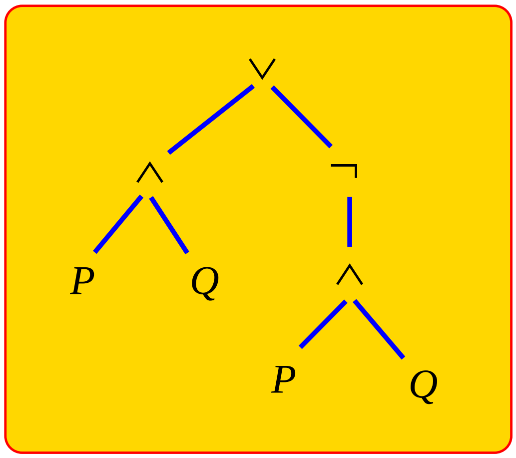
$$P \wedge Q \vee R$$

would be ambiguous.

The **tree** of the formula

$$(P \wedge Q) \vee \neg (P \wedge Q)$$

is given by:



The **subformulas** of $(P \wedge Q) \vee \neg (P \wedge Q)$ are:

$(P \wedge Q) \vee \neg (P \wedge Q)$ $\quad$ $P \wedge Q$

$\neg (P \wedge Q)$ $\quad$ P $\quad$ Q

The **Subformulas** of F (inductive definition):

• The only subformula of a propositional variable P is P itself.

• The only subformula of a constant c is c itself (c is 0 or 1).

• The subformulas of $\neg$F are $\neg$F, and all subformulas of F.

• The subformulas of G $\Box$ H are G $\Box$ H and all subformulas of G and all subformulas of H.

($\Box$ denotes any of $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$.)

If we assign **truth values** to the variables in a propositional formula then we can calculate the truth value of the formula.

This is based on the **truth tables for the connectives**:

not

P	$\neg P$
1	0
0	1

and

P	Q	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

or

P	Q	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0

implies

P	Q	$P \rightarrow Q$
1	1	1
1	0	0
0	1	1
0	0	1

iff

P	Q	$P \leftrightarrow Q$
1	1	1
1	0	0
0	1	0
0	0	1

Now, given any propositional formula F we have a **truth table** for F.

For $(P \vee Q) \rightarrow (P \leftrightarrow Q)$ we have

P	Q	$(P \vee Q) \rightarrow (P \leftrightarrow Q)$
1	1	1
1	0	0
0	1	0
0	0	1

A longer version of the truth table includes the truth tables for the subformulas:

P	Q	$P \vee Q$	$P \leftrightarrow Q$	$(P \vee Q) \rightarrow (P \leftrightarrow Q)$
1	1	1	1	1
1	0	1	0	0
0	1	1	0	0
0	0	0	1	1

A **truth evaluation** $\mathbf{e} = (e_1, \ldots, e_n)$ for the list $P_1, \ldots, P_n$ of propositional variables is a sequence of n truth values.

Thus $\mathbf{e} = (1, 1, 0, 1)$ is a truth evaluation for the variables P, Q, R, S.

Given a formula $\mathsf{F}(P_1, \ldots, P_n)$ let $\mathsf{F}(\mathbf{e})$ denote the **propositional formula** $\mathsf{F}(e_1, \ldots, e_n)$.

If the formula has four variables, say $\mathsf{F}(P, Q, R, S)$, then for the $\mathbf{e}$ above we have $\mathsf{F}(\mathbf{e}) = \mathsf{F}(1, 1, 0, 1)$.

Let $\widehat{\mathsf{F}}(\mathbf{e})$ be the **truth value** of F at $\mathbf{e}$.

Example

Let $\mathsf{F}(P, Q, R, S)$ be the formula

$$\neg(P \vee R) \rightarrow (S \wedge Q),$$

and let $\mathbf{e}$ be the truth evaluation $(1, 1, 0, 1)$ for P, Q, R, S.

Then $\mathsf{F}(\mathbf{e})$ is $\neg(1 \vee 0) \rightarrow (1 \wedge 1)$,

and $\widehat{\mathsf{F}}(\mathbf{e}) = 1$.

Equivalent Formulas

F and G are **(truth) equivalent,** written F $\sim$ G, if they have the same truth tables.

Example:

$$\begin{aligned} 1 &\sim P \vee \neg P \\ 0 &\sim \neg(P \vee \neg P) \\ P \wedge Q &\sim \neg(\neg P \vee \neg Q) \\ P \rightarrow Q &\sim \neg P \vee Q \\ P \leftrightarrow Q &\sim \neg(\neg P \vee \neg Q) \vee \neg(P \vee Q). \end{aligned}$$

We have just expressed the standard connectives in terms of $\neg$, $\vee$.

Proving Formulas are Equivalent

P	Q	$P \to Q$	$\neg Q \to \neg P$	$\neg P \vee Q$
1	1	1	1	1
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

P	Q	R	$P \wedge (Q \vee R)$	$(P \wedge Q) \vee (P \wedge R)$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	0	0
0	1	1	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

Fundamental (Truth) Equivalences

1.	$P \vee P$	$\sim$	P	idempotent
2.	$P \wedge P$	$\sim$	P	idempotent
3.	$P \vee Q$	$\sim$	$Q \vee P$	commutative
4.	$P \wedge Q$	$\sim$	$Q \wedge P$	commutative
5.	$P \vee (Q \vee R)$	$\sim$	$(P \vee Q) \vee R$	associative
6.	$P \wedge (Q \wedge R)$	$\sim$	$(P \wedge Q) \wedge R$	associative
7.	$P \wedge (P \vee Q)$	$\sim$	P	absorption
8.	$P \vee (P \wedge Q)$	$\sim$	P	absorption
9.	$P \wedge (Q \vee R)$	$\sim$	$(P \wedge Q) \vee (P \wedge R)$	distributive
10.	$P \vee (Q \wedge R)$	$\sim$	$(P \vee Q) \wedge (P \vee R)$	distributive

11.	$P \vee \neg P$	$\sim$	1	excluded middle
12.	$P \wedge \neg P$	$\sim$	0	
13.	$\neg \neg P$	$\sim$	P	
14.	$P \vee 1$	$\sim$	1	
15.	$P \wedge 1$	$\sim$	P	
16.	$P \vee 0$	$\sim$	P	
17.	$P \wedge 0$	$\sim$	0	
18.	$\neg (P \vee Q)$	$\sim$	$\neg P \wedge \neg Q$	De Morgan's law
19.	$\neg (P \wedge Q)$	$\sim$	$\neg P \vee \neg Q$	De Morgan's law
20.	$P \rightarrow Q$	$\sim$	$\neg P \vee Q$	
21.	$P \rightarrow Q$	$\sim$	$\neg Q \rightarrow \neg P$	

22.	$P \to (Q \to R)$	$\sim$	$(P \wedge Q) \to R$
23.	$P \to (Q \to R)$	$\sim$	$(P \to Q) \to (P \to R)$
24.	$P \leftrightarrow P$	$\sim$	1
25.	$P \leftrightarrow Q$	$\sim$	$Q \leftrightarrow P$
26.	$(P \leftrightarrow Q) \leftrightarrow R$	$\sim$	$P \leftrightarrow (Q \leftrightarrow R)$
27.	$P \leftrightarrow \neg Q$	$\sim$	$\neg (P \leftrightarrow Q)$
28.	$P \leftrightarrow (Q \leftrightarrow P)$	$\sim$	Q
29.	$P \leftrightarrow Q$	$\sim$	$(P \to Q) \wedge (Q \to P)$
30.	$P \leftrightarrow Q$	$\sim$	$(P \wedge Q) \vee (\neg P \wedge \neg Q)$
31.	$P \leftrightarrow Q$	$\sim$	$(P \vee \neg Q) \wedge (\neg P \vee Q)$

A Few More Useful Equivalences

$$1 \leftrightarrow P \quad \sim \quad P$$

$$0 \leftrightarrow P \quad \sim \quad \neg P$$

$$1 \rightarrow P \quad \sim \quad P$$

$$P \rightarrow 1 \quad \sim \quad 1$$

$$0 \rightarrow P \quad \sim \quad 1$$

$$P \rightarrow 0 \quad \sim \quad \neg P$$

Tautologies and Contradictions

F is a **tautology** if $\widehat{F}(\mathbf{e}) = 1$ for every truth evaluation $\mathbf{e}$. This means the truth table for F looks like:

	F
	1
	⋮
	1

Theorem

F and G are truth equivalent iff the formula $F \leftrightarrow G$ is a tautology.

F is a **contradiction** if $\widehat{F}(\mathbf{e}) = 0$ for every truth evaluation $\mathbf{e}$. This means the truth table looks like:

	F
	0
	⋮
	0

Substitution means uniform substitution of formulas for variables.

$\mathsf{F}(\mathsf{H}_1, \ldots, \mathsf{H}_n)$ means:

substitute H_i for each occurrence of P_i in $\mathsf{F}(P_1, \ldots, P_n)$.

Example

Thus if $\mathsf{F}(P, Q)$ is $P \rightarrow (Q \rightarrow P)$ then

$\mathsf{F}(\underline{\neg P \vee R}, \underline{\neg P})$ is $\underline{\neg P \vee R} \rightarrow (\underline{\neg P} \rightarrow \underline{\neg P \vee R})$.

Substitution Theorem

From

$$\mathsf{F}(P_1, \ldots, P_n) \ \sim \ \mathsf{G}(P_1, \ldots, P_n),$$

we can conclude

$$\mathsf{F}(\mathsf{H}_1, \ldots, \mathsf{H}_n) \ \sim \ \mathsf{G}(\mathsf{H}_1, \ldots, \mathsf{H}_n).$$

Example

From the DeMorgan law

$$\neg (P \vee Q) \ \sim \ \neg P \wedge \neg Q$$

we have:

$$\neg (\underline{(P \rightarrow R)} \vee \underline{(R \leftrightarrow Q)}) \ \sim \ \neg \underline{(P \rightarrow R)} \wedge \neg \underline{(R \leftrightarrow Q)}.$$

[Some Exercises]

Which of the following propositional formulas are substitution instances of the formula

$P \rightarrow (Q \rightarrow P)$?

If a formula is indeed a substitution instance, give the formulas substituted for P, Q.

$\neg R \rightarrow (R \rightarrow \neg R)$

$\neg R \rightarrow (\neg R \rightarrow \neg R)$

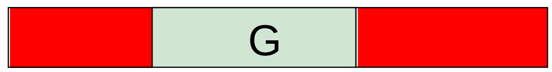
$\neg R \rightarrow (\neg R \rightarrow R)$

$(P \wedge Q \rightarrow P) \rightarrow ((Q \rightarrow P) \rightarrow (P \wedge Q \rightarrow P))$

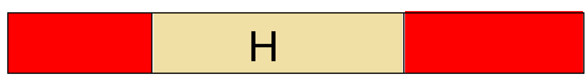
$((P \rightarrow P) \rightarrow P) \rightarrow ((P \rightarrow (P \rightarrow (P \rightarrow P))))$?

Replacement

If F has a subformula G, say

F =


then, when we **replace** the given occurrence of G by another formula H, the result looks like

F′ =


Some like to call this substitution as well. But then there are two kinds of substitution! For clarity it is better to call it replacement.

Example

If we replace the second occurrence of $P \vee Q$ in the formula F

$$(P \vee Q) \rightarrow (R \leftrightarrow (P \vee Q))$$

by the formula $Q \vee P$ then we obtain the formula F′

$$(P \vee Q) \rightarrow (R \leftrightarrow (Q \vee P))$$

Replacement Theorem

From $G \sim H$

we can conclude $F(\cdots G \cdots) \sim F(\cdots H \cdots)$.

Example

From

$$\neg (Q \vee R) \sim \neg Q \wedge \neg R$$

we obtain, by replacement,

$$(P \rightarrow \underline{\neg (Q \vee R)}) \wedge \neg Q \sim (P \rightarrow \underline{\neg Q \wedge \neg R}) \wedge \neg Q$$

Simplification

Simplify the formula

$$(P \wedge Q) \vee \neg(\neg P \vee Q).$$

Solution:

$$\begin{aligned}(P \wedge Q) \vee \neg(\neg P \vee Q) \quad &\sim \quad (P \wedge Q) \vee (\neg\neg P \wedge \neg Q)\\ &\sim \quad (P \wedge Q) \vee (P \wedge \neg Q)\\ &\sim \quad P \wedge (Q \vee \neg Q)\\ &\sim \quad P \wedge 1\\ &\sim \quad P\end{aligned}$$

Because $\sim$ is an equivalence relation we have

$$(P \wedge Q) \vee \neg(\neg P \vee Q) \ \sim \ P$$

Adequate Set of Connectives

Means: Every truth table is the truth table of some propositional formula using the given set of connectives.

The standard connectives are adequate.

Example Find F(P, Q, R) such that

P	Q	R	F
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	1

Answer:

$(P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee$

$(\neg P \wedge Q \wedge R) \vee (\neg P \wedge \neg Q \wedge \neg R)$

Since we only need the connectives $\vee, \wedge, \neg$ to make a formula for any given table it follows that the set of connectives

$$\{\vee, \wedge, \neg\}$$

is adequate.

From the DeMorgan Laws we have

$$P \vee Q \ \sim\ \neg(\neg P \wedge \neg Q)$$
$$P \wedge Q \ \sim\ \neg(\neg P \vee \neg Q)$$

so we see that **both**

$$\{\wedge, \neg\} \quad \text{and} \quad \{\vee, \neg\}$$

are adequate sets of connectives.

And there are other pairs of standard connectives, such as $\{\neg, \rightarrow\}$ (see p. 42 of LMCS), that are adequate.

But **no single standard connective is adequate**.

How can we show this?

The strategy is to show that for each standard connective there is some other standard connective that cannot be expressed using the first standard connective.

If we have a single constant 0 or 1 then we cannot express $\neg$.

If we have just $\neg$ we cannot express $\wedge$.

And for each of the standard binary connectives $\Box$ we claim that it is not possible to express negation.

That is, it is not possible to find a formula $\mathsf{F}(P)$ using just the connective $\Box$ that is equivalent to $\neg P$.

What can we express with $\mathsf{F}(P)$ using only a single connective $\Box$?

For $\square$ =	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
F(P) $\sim$	P	P	1 or P	1 or P

For example, if we start with $\rightarrow$, then any formula F(P) in one variable P, using just the connective $\rightarrow$, is equivalent to either 1 or P.

We prove this using a variation on **Induction**.

The simplest formula F(P) is just the variable P, and P is equivalent to P.

We also have a formula F(P) that is $P \rightarrow P$. This is equivalent to 1.

Can we find a F(P) that is not equivalent to P or 1?

Suppose F(P) is a smallest possible formula that is " equivalent to something else".

Then F(P) must look like G(P) $\rightarrow$ H(P).

But then G(P) and H(P) are too small to be "equivalent to something else".

So they are each equivalent to one of P or 1.

This gives four cases to consider:

$\mathsf{G}(P)$	$\mathsf{H}(P)$	$\mathsf{F}(P) \ = \ \mathsf{G}(P) \rightarrow \mathsf{H}(P)$
P	P	1
P	1	1
1	P	P
1	1	1

So we see that $F(P)$ cannot be "something else".

In particular no $F(P)$ can be $\neg P$.

Single Binary Connectives that are Adequate

The first, $\curlywedge$, was found by Schröder in 1880:

P	Q	$P \curlywedge Q$
1	1	0
1	0	0
0	1	0
0	0	1

Then

$$
\begin{aligned}
1 \quad &\sim \quad (P \curlywedge (P \curlywedge P)) \curlywedge (P \curlywedge (P \curlywedge P)) \\
0 \quad &\sim \quad P \curlywedge (P \curlywedge P) \\
\neg P \quad &\sim \quad P \curlywedge P \\
&\mathit{etc.} \\
P \leftrightarrow Q \quad &\sim \quad ((P \curlywedge P) \curlywedge Q) \curlywedge ((Q \curlywedge P) \curlywedge P).
\end{aligned}
$$

The second, |, was found by Sheffer in 1913, and is the famous **Sheffer stroke**:

P	Q	$P\|Q$
1	1	0
1	0	1
0	1	1
0	0	1

To show that this connective is adequate all we need to do is to **express a known adequate set of connectives in terms of it**.

For the adequate set $\{\neg, \vee\}$ we have:

$$\neg P \sim P|P$$

$$P \vee Q \sim (P|P)|(Q|Q)$$

Associativity and Parentheses

Since the associative law holds for $\vee$ and $\wedge$ it is common practice to drop parentheses in situations such as

$$P \wedge ((Q \wedge R) \wedge S),$$

yielding

$$P \wedge Q \wedge R \wedge S.$$

Likewise we like to write

$$P \vee Q \vee R \vee S,$$

instead of

$$(P \vee Q) \vee (R \vee S).$$

Disjunctive and Conjunctive Forms

Any formula F can be transformed into a **disjunctive form**, e.g.,

$$P \leftrightarrow Q \;\sim\; (P \wedge Q) \vee (\neg P \wedge \neg Q).$$

If each variable or its negation appears in each conjunction then we call it a **disjunctive normal form**. Such conjunctions are **DNF-constituents**.

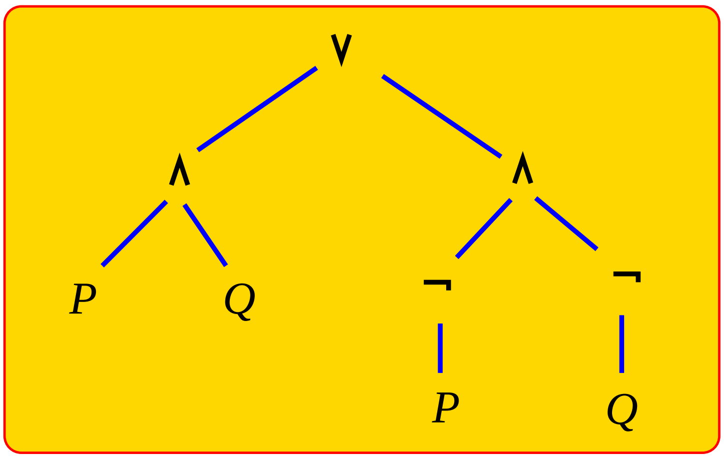
The above disjunctive form is actually a disjunctive normal form, with the DNF-constituents

$$P \wedge Q \quad \text{and} \quad \neg P \wedge \neg Q.$$

The formula tree for the right-hand side

$$(P \wedge Q) \vee (\neg P \wedge \neg Q)$$

is given by:



Notice that the negations are all next to the leaves of the tree.

And there is no $\wedge$ above a $\vee$.

Being in disjunctive form really means:

- negations only appear next to variables, and
- no $\wedge$ is above a $\vee$.

So we can have **degenerate** cases of the disjunctive form:

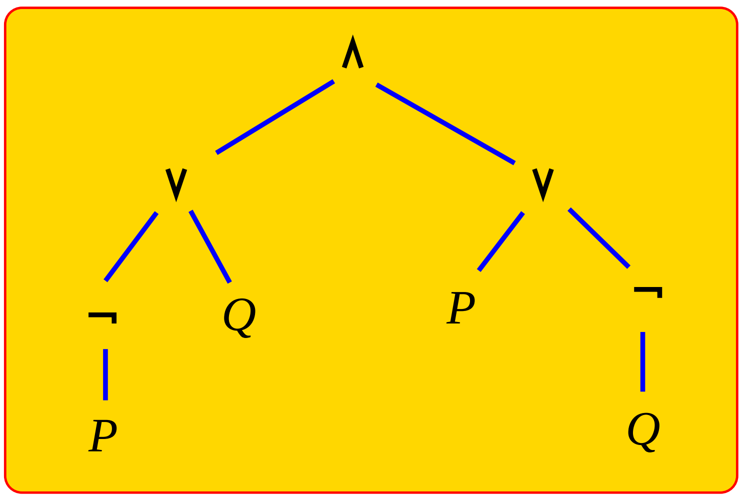
Here are three examples:

$$P \qquad P \vee \neg Q \qquad P \wedge \neg Q$$

And we have **conjunctive forms** such as

$$P \leftrightarrow Q \ \sim \ (\neg P \vee Q) \wedge (P \vee \neg Q).$$

The formula tree for the right-hand side is given by:



Being in conjunctive form means:

- negations only appear next to variables, and
- no $\vee$ is above a $\wedge$.

Simple Cases:

$$F(P, Q) \ = \ P \vee \neg Q$$

is in both disjunctive and conjunctive form.

It is in conjunctive normal form, but not in disjunctive normal form.

The set of variables used affects the normal forms.

Let F(P, Q) be $\neg P$. The normal forms (with respect to P, Q) are:

DNF: $(\neg P \wedge Q) \vee (\neg P \wedge \neg Q)$

CNF: $(\neg P \vee Q) \wedge (\neg P \vee \neg Q)$

Rewrite Rules to Obtain Normal Forms

To transform F into a disjunctive form apply the following:

$$
\begin{aligned}
F \rightarrow G &\rightsquigarrow \neg F \vee G \\
F \leftrightarrow G &\rightsquigarrow (F \rightarrow G) \wedge (G \rightarrow F) \\
\neg (F \vee G) &\rightsquigarrow \neg F \wedge \neg G \\
\neg (F \wedge G) &\rightsquigarrow \neg F \vee \neg G \\
\neg \neg F &\rightsquigarrow F \\
F \wedge (G \vee H) &\rightsquigarrow (F \wedge G) \vee (F \wedge H) \\
(F \vee G) \wedge H &\rightsquigarrow (F \wedge H) \vee (G \wedge H).
\end{aligned}
$$

These rules are applied until no further applications are possible.

Example

$$\begin{aligned} P \wedge (P \rightarrow Q) &\rightsquigarrow P \wedge (\neg P \vee Q) \\ &\rightsquigarrow (P \wedge \neg P) \vee (P \wedge Q) \end{aligned}$$

Now this formula clearly gives a disjunctive form, but not a normal form. We can simplify it considerably, but to do this we need to invoke additional rewrite rules.

More Rewrite Rules:

$$\begin{aligned} 0 \wedge \mathsf{F} &\rightsquigarrow 0 \\ \neg 1 &\rightsquigarrow 0 \\ &etc. \\ \cdots \wedge \mathsf{F} \wedge \cdots \wedge \neg \mathsf{F} \wedge \cdots &\rightsquigarrow 0 \\ \cdots \wedge \mathsf{F} \wedge \cdots \wedge \mathsf{F} \wedge \cdots &\rightsquigarrow \cdots \wedge \mathsf{F} \wedge \cdots \end{aligned}$$

Applying these additional rewrite rules we have:

$$\begin{aligned}(P \wedge \neg P) \vee (P \wedge Q) &\rightsquigarrow 0 \vee (P \wedge Q)\\ &\rightsquigarrow P \wedge Q\end{aligned}$$

One more rule is needed, to handle the exceptional case that the above rules reduce the formula to simply the constant 1.

In this case we rewrite 1 as a join of *all* the DNF–constituents.

Sometimes, after applying all these rules, one still doesn't have a disjunctive normal form.

Example

If we start with $(P \wedge Q) \vee \neg P$ then none of the rules apply.

To get a DNF we need to replace $\neg P$ with $(\neg P \wedge Q) \vee (\neg P \wedge \neg Q)$.

Then

$$(P \wedge Q) \vee \neg P \quad \sim \quad (P \wedge Q) \vee (\neg P \wedge Q) \vee (\neg P \wedge \neg Q).$$

Now we have a disjunctive normal form.

The second method to find normal forms is to **use truth tables**.

The rows of the truth table of F yield the constituents of F according to

• the DNF–constituents are in 1–1 correspondence with the lines of the truth table for which F is **true**.

• The CNF–constituents are in 1–1 correspondence with the lines of the truth table for which F is **false**.

Example

The DNF– and CNF–constituents for the rows of a truth table in the variables P, Q, R:

P	Q	R	DNF–constituent	CNF–constituent
1	1	1	$P \wedge Q \wedge R$	$\neg P \vee \neg Q \vee \neg R$
1	1	0	$P \wedge Q \wedge \neg R$	$\neg P \vee \neg Q \vee R$
etc.				
0	0	0	$\neg P \wedge \neg Q \wedge \neg R$	$P \vee Q \vee R$

A DNF constitutent is true only for its row.

A CNF constitutent is false only for its row.

Example

Using truth tables to find the **disjunctive normal form**.

P	Q	$(\neg P \vee Q) \wedge \neg P$
1	1	0
1	0	0
0	1	1
0	0	1

The disjunctive normal form for

$$(\neg P \wedge Q) \vee \neg P$$

is

$$(\neg P \wedge Q) \vee (\neg P \wedge \neg Q).$$

Example

Using truth tables to find the

conjunctive normal form:

P	Q	$(P \leftrightarrow Q) \vee (P \rightarrow \neg Q)$
1	1	1
1	0	0
0	1	1
0	0	1

The conjunctive normal form for

$$(P \leftrightarrow Q) \vee (P \rightarrow \neg Q)$$

is

$$\neg P \vee Q.$$

Unique Normal Forms

A formula has many disjunctive forms, and many conjunctive forms.

But it has **only one disjunctive normal form**.

And **only one conjunctive normal form**.

(Since normal forms are determined by the truth table of a formula.)

Two formulas are equivalent iff they have the same disjunctive (or conjunctive) normal forms.

A **(logical) argument** draws conclusions from premisses.

What constitutes a valid argument?

Definition An argument $\mathsf{F}_1, \cdots, \mathsf{F}_n \therefore \mathsf{F}$ is **valid** (or **correct**) provided:

the conclusion is true whenever the premisses are true, i.e.,

	F_1	$\cdots$	F_n	F
$\mathbf{e}$	1	$\cdots$	1	

implies

	F_1	$\cdots$	F_n	F
$\mathbf{e}$	1	$\cdots$	1	1

Proposition

$$\mathsf{F}_1, \cdots, \mathsf{F}_n \quad \therefore \mathsf{F}$$

is valid iff

$$\mathsf{F}_1 \wedge \cdots \wedge \mathsf{F}_n \rightarrow \mathsf{F}$$

is a tautology.

Both of these say that F is true whenever $\mathsf{F}_1, \ldots, \mathsf{F}_n$ are true.

Example

(Chrysippus: A good hunting dog has basic skills in reasoning.)

> When running after a rabbit, the dog found that the path suddenly split in three directions.
>
> The dog sniffed the first path and found no scent;
>
> then it sniffed the second path and found no scent;
>
> then, without bothering to sniff the third path, it ran down that path.

We can summarize the canine's fine reasoning as follows:

- The rabbit went this way or that way or the other way.

- Not this way.

- Not that way.

- Therefore the other way.

We can express the argument as

$$\begin{array}{c} P \vee Q \vee R \\ \neg P \\ \neg Q \\ \therefore R. \end{array}$$

For the argument

$$\begin{array}{c} P \vee Q \vee R \\ \neg P \\ \neg Q \\ \therefore R \end{array}$$

the validity can easily be checked using a truth table:

P	Q	R	$P \vee Q \vee R$	$\neg P$	$\neg Q$	R
1	1	1	1	0	0	1
1	1	0	1	0	0	0
1	0	1	1	0	1	1
1	0	0	1	0	1	0
0	1	1	1	1	0	1
0	1	0	1	1	0	0
0	0	1	1	1	1	1
0	0	0	0	1	1	0

Satisfiable

A set $\mathcal{S}$ of propositional formulas is **satisfiable** if there is a truth evaluation $\mathbf{e}$ for the variables in $\mathcal{S}$ that makes every formula in $\mathcal{S}$ true.

We say that $\mathbf{e}$ **satisfies** $\mathcal{S}$.

The expression $\mathsf{Sat}(\mathcal{S})$ means that $\mathcal{S}$ **is satisfiable**;

the expression $\neg\,\mathsf{Sat}(\mathcal{S})$ means that $\mathcal{S}$ **is not satisfiable**.

Thus a finite set $\{\mathsf{F}_1, \ldots, \mathsf{F}_n\}$ of formulas is satisfiable iff when we look at the combined truth table for the F_i we can find a line that looks as follows:

P_1	$\cdots$	P_m	F_1	$\cdots$	F_n
e_1	$\cdots$	e_m	1	$\cdots$	1

Example

Let $\mathcal{S}$ be the set of formulas

$$\{P \to Q,\ Q \to R,\ R \to P\}.$$

The combined truth table is:

	P	Q	R	$P \to Q$	$Q \to R$	$R \to P$
1.	1	1	1	1	1	1
2.	1	1	0	1	0	1
3.	1	0	1	0	1	1
4.	1	0	0	0	1	1
5.	0	1	1	1	1	0
6.	0	1	0	1	0	1
7.	0	0	1	1	1	0
8.	0	0	0	1	1	1

Thus $\mathcal{S}$ is satisfiable.

Example

Let $\mathcal{S}$ be the set of formulas

$$\{P \leftrightarrow \neg Q,\ Q \leftrightarrow R,\ R \leftrightarrow P\}.$$

The combined truth table is:

	P	Q	R	$P \leftrightarrow \neg Q$	$Q \leftrightarrow R$	$R \leftrightarrow P$
1.	1	1	1	0	1	1
2.	1	1	0	0	0	0
3.	1	0	1	1	0	1
4.	1	0	0	1	1	0
5.	0	1	1	1	1	0
6.	0	1	0	1	0	1
7.	0	0	1	0	0	0
8.	0	0	0	0	1	1

Thus $\mathcal{S}$ is **not** satisfiable.

Valid Arguments
and
Not Satisfiable Formulas

The following assertions are equivalent:

- $\mathsf{F}_1, \cdots, \mathsf{F}_n \quad \therefore \mathsf{F}$ is valid.

- $\{\mathsf{F}_1, \cdots, \mathsf{F}_n, \neg \mathsf{F}\}$ is not satisfiable.

- $\mathsf{F}_1 \wedge \cdots \wedge \mathsf{F}_n \ \rightarrow \ \mathsf{F}$ is a tautology.

- $\mathsf{F}_1 \wedge \cdots \wedge \mathsf{F}_n \wedge \neg \mathsf{F}$ is not satisfiable.

Each of these says that F is true whenever $\mathsf{F}_1, \ldots, \mathsf{F}_n$ are true.

From a combined truth table such as

	P	Q	R	F1	F2	F3	F4
1.	1	1	1	1	1	0	1
2.	1	1	0	0	0	1	0
3.	1	0	1	1	1	0	0
4.	1	0	0	0	0	0	0
5.	0	1	1	1	0	0	0
6.	0	1	0	0	0	1	1
7.	0	0	1	1	1	0	1
8.	0	0	0	0	0	0	1

we can read off information about

- **normal forms**
- **tautologies**
- **satisfiable**
- **equivalence**
- **contradictions**
- **valid arguments**

Example (translation into propositional logic)

1. Good-natured tenured mathematics professors are dynamic.

$$A \wedge B \wedge C \rightarrow D$$

2. Grumpy student advisors play slot machines.

$$\neg A \wedge M \rightarrow L$$

A Tufa Problem

The island of Tufa has two tribes, the Tu's who always tell the truth, and the Fa's who always lie.

A traveler encountered three residents A,B, and C of Tufa, and each made a statement to the traveler:

A said, " A or B tells the truth if C lies."

B said, "If A or C tell the truth, then it is not the case that exactly one of us is telling the truth."

C said, " A or B is lying iff A or C is telling the truth."

Determine, as best possible, which tribes A, B, and C belong to?

Let A be the statement "A is telling the truth" (and thus A is a Tu), etc.

Then in symbolic form we have:

A says: $\neg C \to (A \vee B)$

B says: $A \vee C \to$
$\neg((\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C))$

C says: $\neg(A \wedge B) \leftrightarrow (A \vee C)$.

The following statements are given to be true:

$A \leftrightarrow (\neg C \rightarrow (A \vee B))$

$B \leftrightarrow (A \vee C \rightarrow \neg((\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C)))$

$C \leftrightarrow (\neg (A \wedge B) \leftrightarrow (A \vee C))$.

Letting these three propositional formulas be F, G, and H, we have the combined truth table:

	A	B	C	F	G	H
1.	1	1	1	1	1	0
2.	1	1	0	1	1	1
3.	1	0	1	1	0	0
4.	1	0	0	1	1	1
5.	0	1	1	0	1	0
6.	0	1	0	0	1	0
7.	0	0	1	0	1	1
8.	0	0	0	1	0	1

From lines 2 and 4 we see that A must be a Tu and C must be a Fa.

We do not know which tribe B belongs to.

The FŁ Propositional Logic

Propositional Variables: $P, Q, \ldots$

Connectives: $\neg\,, \rightarrow$

Rule of inference: $\dfrac{\mathsf{F},\ \ \mathsf{F} \rightarrow \mathsf{G}}{\mathsf{G}}$

(modus ponens)

Axiom schemata:

A1: $\mathsf{F} \rightarrow (\mathsf{G} \rightarrow \mathsf{F})$

A2: $(\mathsf{F} \rightarrow (\mathsf{G} \rightarrow \mathsf{H})) \rightarrow ((\mathsf{F} \rightarrow \mathsf{G}) \rightarrow (\mathsf{F} \rightarrow \mathsf{H}))$

A3: $(\neg\,\mathsf{F} \rightarrow \neg\,\mathsf{G}) \rightarrow (\mathsf{G} \rightarrow \mathsf{F})$

$\mathcal{S} \vdash$ F [read: F **can be derived from** $\mathcal{S}$] means there is a sequence of formulas

$$\mathsf{F}_1, \ldots, \mathsf{F}_n \ ,$$

with F $=$ F_n, such that for each i

- either F_i is an axiom,
- or F_i is in $\mathcal{S}$,
- or F_i is obtained from two previous F_js by an application of modus ponens.

$\mathsf{F}_1, \ldots, \mathsf{F}_n$ is an $\mathcal{S}$–**derivation** (or $\mathcal{S}$–**proof**) of F.

A ∅–derivation is simply called a **derivation**.

Note: The axioms are tautologies.

The proof system is **sound**.

(If $\vdash F$ then F is a tautology.)

Now we start to prove **completeness**.

(If F is a tautology then $\vdash F$.)

First two lemmas that are not in the text.

Lemma A

If F is an axiom then $\mathcal{S} \vdash F$.

Lemma B

If $F \in \mathcal{S}$ then $\mathcal{S} \vdash F$.

Lemma D.0.5 $\vdash \mathsf{F} \rightarrow \mathsf{F}$.

Proof

1.	$\mathsf{F} \rightarrow ((\mathsf{F} \rightarrow \mathsf{F}) \rightarrow \mathsf{F})$	A1
2.	$(\mathsf{F} \rightarrow ((\mathsf{F} \rightarrow \mathsf{F}) \rightarrow \mathsf{F})) \rightarrow ((\mathsf{F} \rightarrow (\mathsf{F} \rightarrow \mathsf{F})) \rightarrow (\mathsf{F} \rightarrow \mathsf{F}))$	A2
3.	$(\mathsf{F} \rightarrow (\mathsf{F} \rightarrow \mathsf{F})) \rightarrow (\mathsf{F} \rightarrow \mathsf{F})$	MP 1,2
4.	$\mathsf{F} \rightarrow (\mathsf{F} \rightarrow \mathsf{F})$	A1
5.	$\mathsf{F} \rightarrow \mathsf{F}.$	MP 3,4

Lemma D.0.6

If $\mathcal{S} \vdash \mathsf{F}$ and $\mathcal{S} \vdash \mathsf{F} \rightarrow \mathsf{G}$, then $\mathcal{S} \vdash \mathsf{G}$.

Proof

Let $\mathsf{F}_1, \ldots, \mathsf{F}_m$ be an $\mathcal{S}$–derivation of F, and let $\mathsf{F}_{m+1}, \ldots, \mathsf{F}_n$ be an $\mathcal{S}$–derivation of $\mathsf{F} \rightarrow \mathsf{G}$.

Then $\mathsf{F}_1, \ldots, \mathsf{F}_n, \mathsf{G}$ is an $\mathcal{S}$–derivation of G.

Lemma D.0.7

If $\mathcal{S} \vdash \mathsf{F}$ and $\mathcal{S} \subseteq \mathcal{S}'$, then $\mathcal{S}' \vdash \mathsf{F}$.

Proof

Let $\mathsf{F}_1, \ldots, \mathsf{F}_n$ be an $\mathcal{S}$–derivation of F.
Then it is also an $\mathcal{S}'$–derivation of F.

Lemma D.0.8
[Deduction Lemma / Herbrand 1930]

$$\mathcal{S} \cup \{\mathsf{F}\} \vdash \mathsf{G} \qquad \text{iff} \qquad \mathcal{S} \vdash \mathsf{F} \rightarrow \mathsf{G}.$$

Proof

The direction $(\Leftarrow)$ follows from taking a derivation $\mathsf{F}_1, \ldots, \mathsf{F}_n$ of $\mathsf{F} \rightarrow \mathsf{G}$ from $\mathcal{S}$ and attaching the two formulas F, G to the end of it.

The other direction follows by an **induction proof** on the length of a derivation of G from $\mathcal{S} \cup \{\mathsf{F}\}$. (See the text.)

Lemma D.0.9

If $\mathcal{S} \vdash F \to G$ and $\mathcal{S} \vdash G \to H$, then $\mathcal{S} \vdash F \to H$.

Proof

1.	$\mathcal{S}$	$\vdash$	$F \to G$	given
2.	$\mathcal{S}$	$\vdash$	$G \to H$	given
3.	$\mathcal{S} \cup \{F\}$	$\vdash$	G	1 D.0.8
4.	$\mathcal{S} \cup \{F\}$	$\vdash$	$G \to H$	2 D.0.7
5.	$\mathcal{S} \cup \{F\}$	$\vdash$	H	3,4 D.0.6
6.	$\mathcal{S}$	$\vdash$	$F \to H.$	5 D.0.8

A list of other lemmas needed:

Lemma D.0.10 If $\mathcal{S} \vdash \mathsf{F} \rightarrow (\mathsf{G} \rightarrow \mathsf{H})$ and $\mathcal{S} \vdash \mathsf{G}$, then $\mathcal{S} \vdash \mathsf{F} \rightarrow \mathsf{H}$.

Lemma D.0.11 $\vdash \neg\,\mathsf{F} \rightarrow (\mathsf{F} \rightarrow \mathsf{G})$.

Lemma D.0.12 $\vdash \neg\,\neg\,\mathsf{F} \rightarrow \mathsf{F}$.

Lemma D.0.13 $\vdash \mathsf{F} \rightarrow \neg\,\neg\,\mathsf{F}$.

Lemma D.0.14 $\vdash (\mathsf{F} \rightarrow \mathsf{G}) \rightarrow (\neg\,\mathsf{G} \rightarrow \neg\,\mathsf{F})$.

Lemma D.0.15 $\vdash \mathsf{F} \rightarrow (\neg\,\mathsf{G} \rightarrow \neg\,(\mathsf{F} \rightarrow \mathsf{G}))$.

Lemma D.0.16 If $\mathcal{S} \cup \{\mathsf{F}\} \vdash \mathsf{G}$ and $\mathcal{S} \cup \{\neg\,\mathsf{F}\} \vdash \mathsf{G}$, then $\mathcal{S} \vdash \mathsf{G}$.

Let $\mathsf{F}(P_1, \ldots, P_n)$ be a propositional formula.

Let $\widetilde{P}_1, \ldots, \widetilde{P}_n$ be such that $\widetilde{P}_i \in \{P_i, \neg P_i\}$.

Let $\mathbf{e}$ be a truth evaluation such that
$\mathbf{e}(\widetilde{P}_i) = 1$, for $i \leq n$.

Then let $$\widetilde{\mathsf{F}} = \begin{cases} \mathsf{F} & \text{if } \mathbf{e}(\mathsf{F}) = 1 \\ \neg \mathsf{F} & \text{if } \mathbf{e}(\mathsf{F}) = 0. \end{cases}$$

Lemma [Kalmar]
Let F, $\widetilde{P}_1, \ldots, \widetilde{P}_n$, $\widetilde{\mathsf{F}}$ and $\mathbf{e}$ be above. Then

$$\widetilde{P}_1, \ldots, \widetilde{P}_n \vdash \widetilde{\mathsf{F}}.$$

Proof. (See Text.)

Theorem [Completeness]

$$\vdash \mathsf{F} \quad \text{if} \quad \mathsf{F} \text{ is a tautology.}$$

Proof

Let $\mathsf{F}(P_1, \ldots, P_n)$ be a tautology. Then for any $\widetilde{P}_1, \ldots, \widetilde{P}_n$ we have $\widetilde{\mathsf{F}} = \mathsf{F}$.

Thus

1.	$\widetilde{P}_1, \ldots, \widetilde{P}_{n-1}, P_n \vdash \mathsf{F}$	Kalmar
2.	$\widetilde{P}_1, \ldots, \widetilde{P}_{n-1}, \neg P_n \vdash \mathsf{F}$	Kalmar
3.	$\widetilde{P}_1, \ldots, \widetilde{P}_{n-1} \vdash \mathsf{F}$	D.0.16

Continuing, we have $\vdash \mathsf{F}$.

Resolution is a rule of inference used to show a set of propositional formulas of the form

$$\widetilde{P}_1 \vee \cdots \vee \widetilde{P}_m \quad (\star)$$

is **not** satisfiable.

Here $\widetilde{P}_i$ means P_i or $\neg P_i$, where P_i is a propositional variable.

A Justification for Using Resolution:

We know that an argument

$$\mathsf{F}_1, \cdots, \mathsf{F}_n \;\therefore \mathsf{F}$$

is valid iff

$$\mathsf{F}_1 \wedge \cdots \wedge \mathsf{F}_n \wedge \neg \mathsf{F} \quad (\star\star)$$

is **not** satisfiable.

If we put each of the F_i as well as $\neg\mathsf{F}$ into **conjunctive** form then we can replace $(\star\star)$ by a collection of formulas of the form $(\star)$.

Thus saying that an argument is valid is equivalent to saying that a certain set of disjunctions of variables and negated variables is not satisfiable.

Example

To determine the validity of the argument

$$P \to Q,\ \neg P \to R,\ Q \vee R \to S \quad \therefore S$$

we consider the satisfiability of

$$\{P \to Q,\ \neg P \to R,\ Q \vee R \to S,\ \neg S\}.$$

Converting this into the desired disjunctions gives

$$\{\neg P \vee Q,\ P \vee R,\ \neg Q \vee S,\ \neg R \vee S,\ \neg S\}.$$

It is easily checked that this is not satisfiable, so the propositional argument is valid.

Literals

P and $\neg P$ are called **literals**.

The **complement** $\overline{\mathsf{L}}$ of a literal L is defined by:

$$\overline{P} = \neg P \qquad \text{and} \qquad \overline{\neg P} = P$$

P is a **positive** literal.

$\neg P$ is a **negative** literal.

Clauses

Finite sets $\{\mathsf{L}_1, \ldots, \mathsf{L}_m\}$ of literals are called **clauses**.

• A clause $\{\mathsf{L}_1, \cdots, \mathsf{L}_n\}$ is **satisfiable** (by $\mathbf{e}$) if $\mathsf{L}_1 \vee \cdots \vee \mathsf{L}_n$ is satisfiable (by $\mathbf{e}$).

• By definition the **empty clause** $\{\ \}$ is not satisfiable.

• A set $\mathcal{S}$ of clauses is **satisfiable** if there is a truth evaluation $\mathbf{e}$ that satisfies each clause in $\mathcal{S}$.

Resolution

Resolution is the following rule of inference:

$$\frac{\mathsf{C} \cup \{\mathsf{L}\},\ \mathsf{D} \cup \{\overline{\mathsf{L}}\}}{\mathsf{C} \cup \mathsf{D}}$$

Example (of resolution):

$$\frac{\{P, \neg Q, R\},\ \{Q, R, \neg S\}}{\{P, R, \neg S\}}$$

Completeness and Soundness of Resolution

A set $\mathcal{S}$ of clauses is not satisfiable iff one can derive the empty clause from $\mathcal{S}$ using resolution.

Example

A resolution derivation of the empty clause from

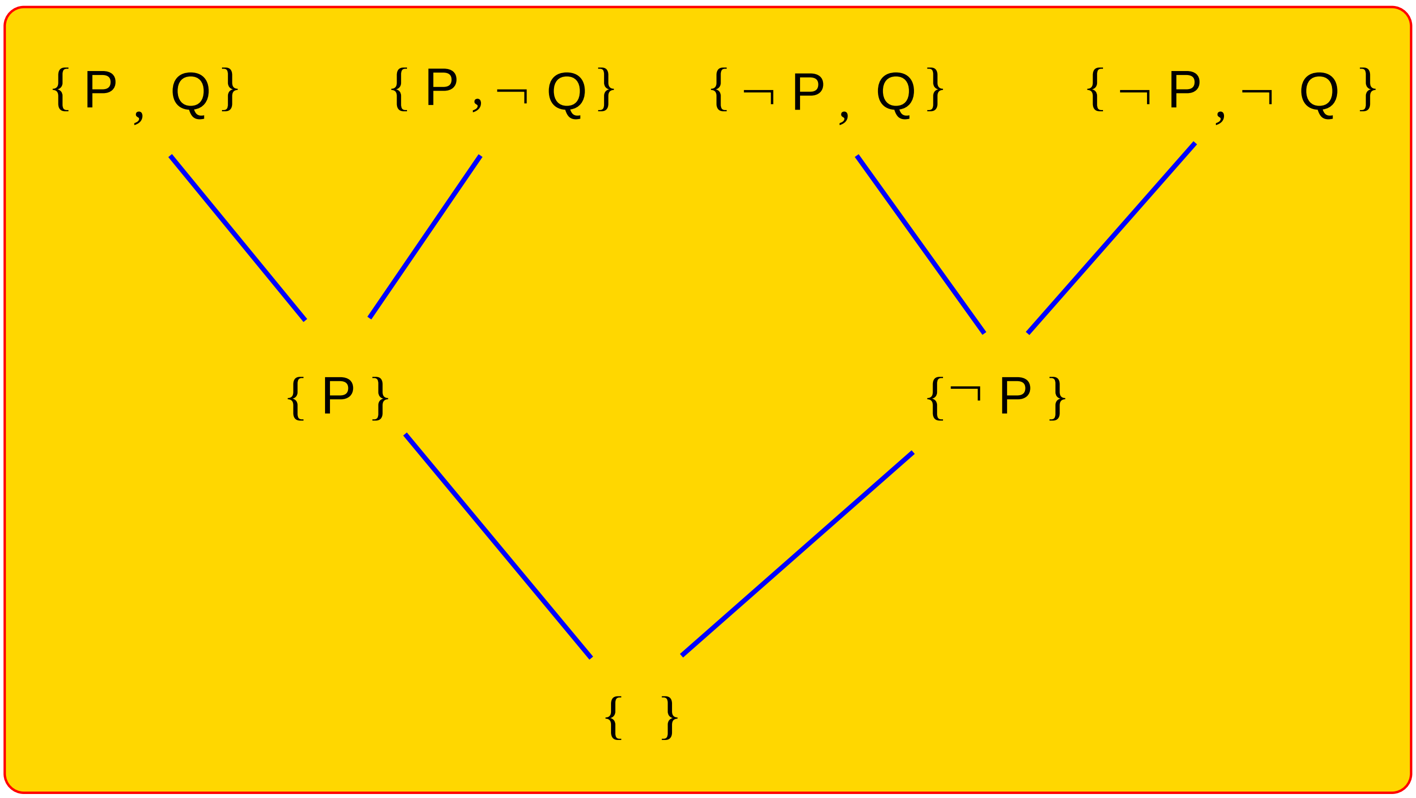
$$\{P, Q\} \quad \{P, \neg Q\} \quad \{\neg P, Q\} \quad \{\neg P, \neg Q\}$$

is

1.	$\{P, Q\}$	given
2.	$\{P, \neg Q\}$	given
3.	$\{P\}$	resolution 1,2
4.	$\{\neg P, Q\}$	given
5.	$\{\neg P, \neg Q\}$	given
6.	$\{\neg P\}$	resolution 4,5
7.	$\{\ \}$	resolution 3,6

This set of clauses is **not satisfiable**.

For the derivation in the previous example we could use the following picture to indicate the resolution steps:



Example

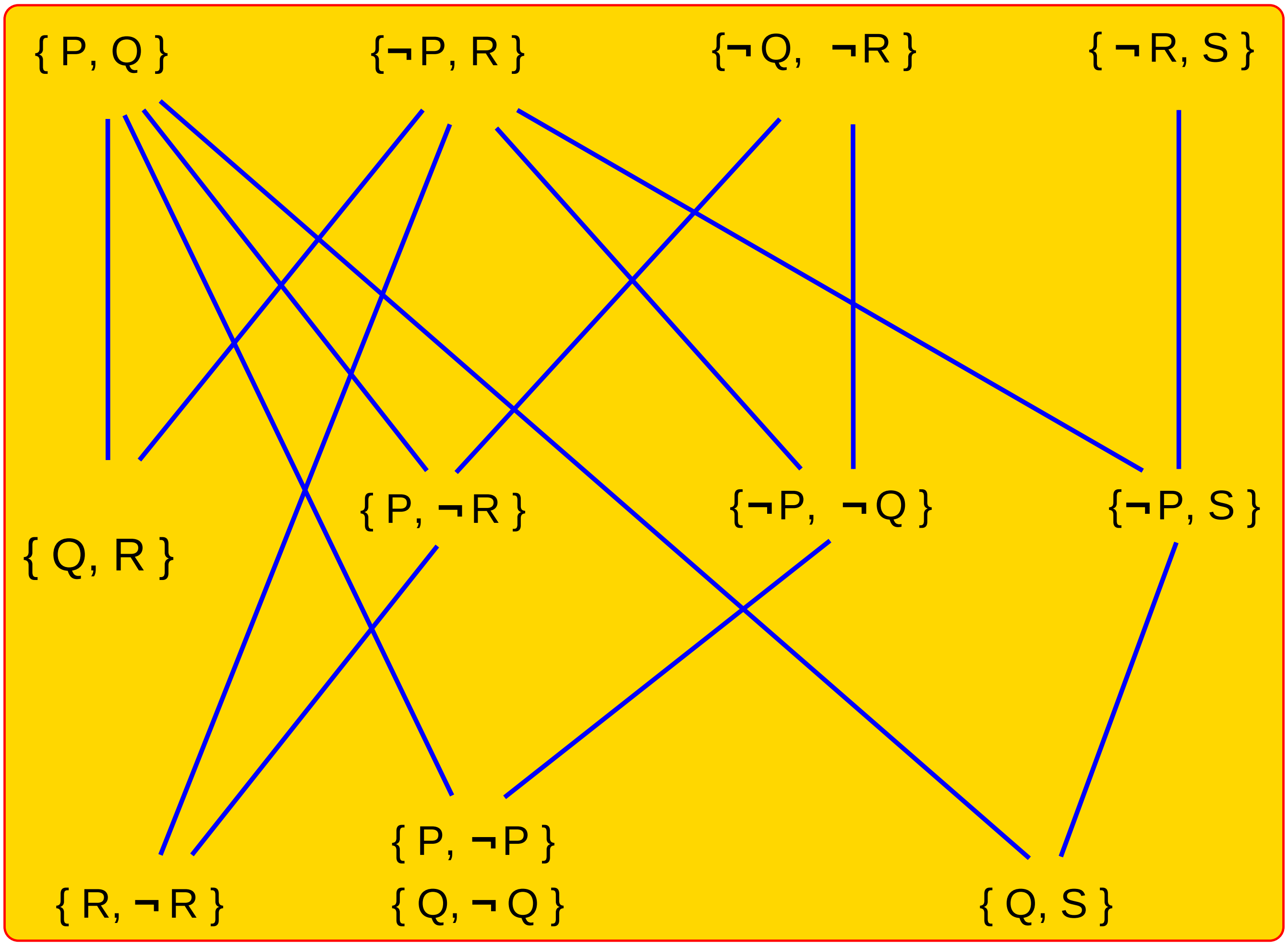
$\mathcal{S} = \{\ \{P, Q\},\ \{\neg P, R\},\ \{\neg Q, \neg R\},\ \{\neg R, S\}\ \}$

1.	$\{P, Q\}$	given	7.	$\{\neg P, \neg Q\}$	2, 3
2.	$\{\neg P, R\}$	given	8.	$\{\neg P, S\}$	2, 4
3.	$\{\neg Q, \neg R\}$	given	9.	$\{Q, \neg Q\}$	1, 7
4.	$\{\neg R, S\}$	given	10.	$\{P, \neg P\}$	1, 7
5.	$\{Q, R\}$	1, 2	11.	$\{Q, S\}$	1, 8
6.	$\{P, \neg R\}$	1, 3	12.	$\{R, \neg R\}$	2, 6

No further clauses can be derived by resolution.

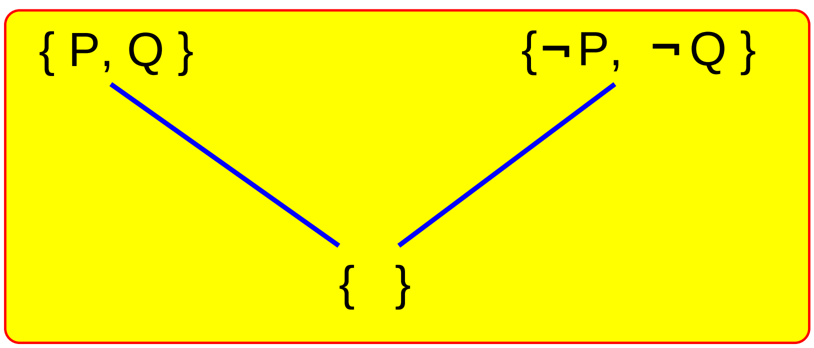
Thus the empty clause cannot be derived, so $\mathcal{S}$ **is satisfiable**.

A diagram of the resolution steps in the previous example:



WARNING

Every term several students will try cancelling two or more complementary literals, as in the following:



This is not correct. Resolution preserves satisfiability!

The Davis-Putnam Procedure (DPP)

- Delete all clauses that are "tautologies".
- Select a variable.
- Add *all* resolvents over that variable.
- Then delete *all* clauses with that variable.

Iterate until there are no variables left.

If you end up with NO CLAUSES left then the original set **is satisfiable**.

Otherwise you must end up with just the EMPTY CLAUSE, and then the original set **is not satisfiable**.

Example

$$\mathcal{S}: \quad \{P, Q\} \quad \{P, \neg Q\} \quad \{\neg P, Q\} \quad \{\neg P, \neg Q\}$$

Eliminating P gives $\{Q\}$, $\{\neg Q\}$, $\{Q, \neg Q\}$.
Eliminating Q gives $\{\ \}$.

So the output is EMPTY CLAUSE.

Example

$\mathcal{S}: \quad \{P, \neg Q\} \quad \{Q\}$.

Eliminating P gives $\{Q\}$.
Eliminating Q gives no clauses.

Thus the output is NO CLAUSES.

Horn Clauses

For certain types of clauses resolution is known to be reasonably fast.

A **Horn clause** is a clause with **at most one positive literal**.

Example

The following is a complete list of the Horn clauses in the three variables P, Q, R:

$$\{\neg P, \neg Q, \neg R\}$$
$$\{P, \neg Q, \neg R\}$$
$$\{\neg P, Q, \neg R\}$$
$$\{\neg P, \neg Q, R\}$$

$\{P, \neg Q, R, \neg S\}$ is not a Horn clause.

Lemma

A resolvent of two Horn clauses is always a Horn clause.

Horn clauses have been popular in **logic programming**, e.g., in Prolog.

Many special kinds of resolution have been developed for Horn clauses—one of the simplest uses unit clauses.

A **unit** clause is a clause $\{L\}$ with a single literal.

Unit resolution

refers to resolution derivations in which at least one of the clauses used in each resolution step is a unit clause.

Theorem
Unit resolution is **sound and complete** for Horn clauses.

That is, a set of Horn clauses is not satisfiable iff one can derive the empty clause using unit resolution.

For resolution with unit clauses the resolvents do not grow in size.

Example

Using unit resolution on Horn clauses with

$\mathcal{S} =$

$\{\ \{P\},\quad \{\neg P, \neg R, S\},\quad \{R\},\quad \{\neg R, \neg S, \neg T\}\ \}$

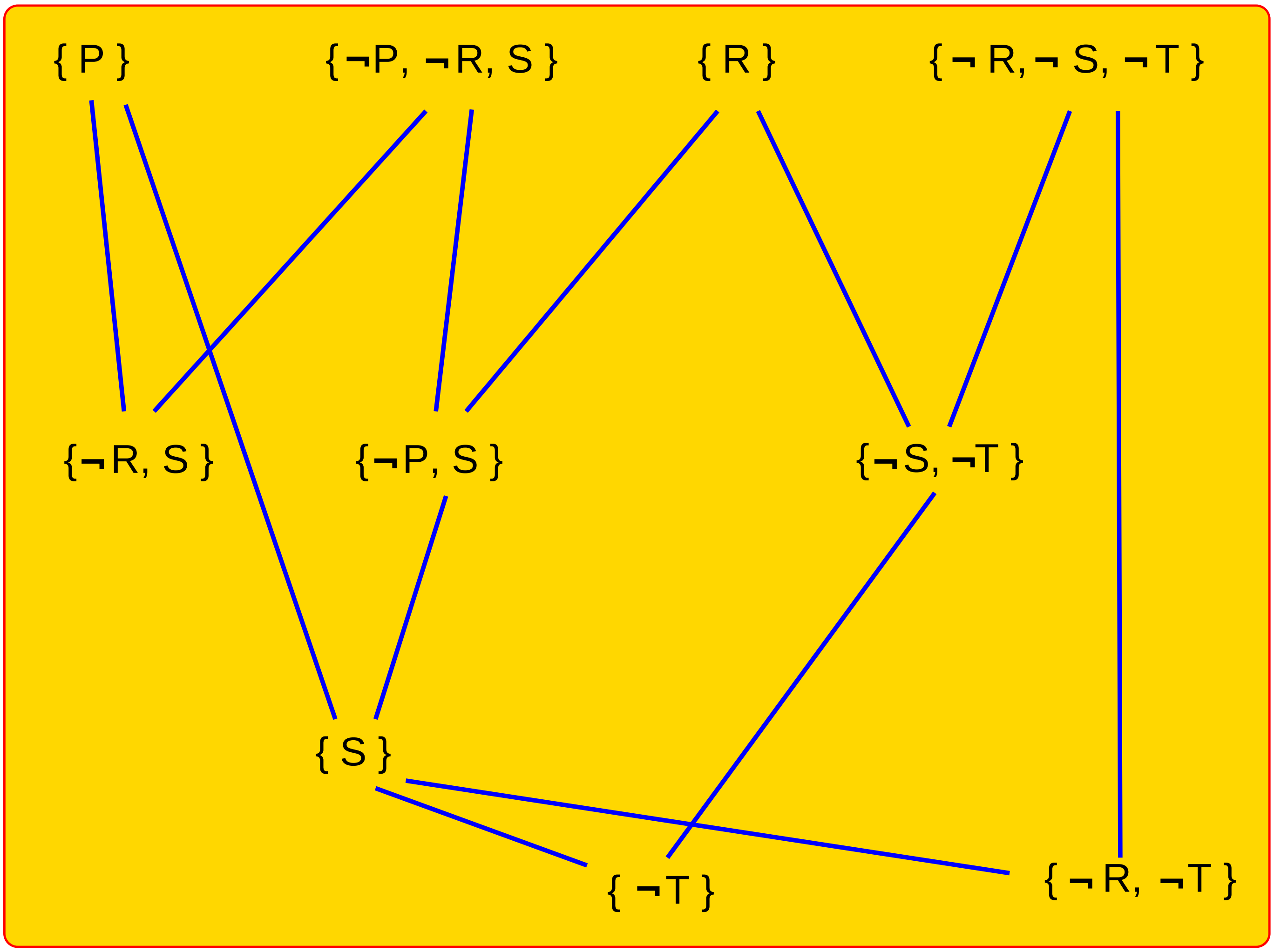
$\mathbf{1}$.	$\{P\}$	given	6.	$\{\neg P, S\}$	$2, \mathbf{3}$
2.	$\{\neg P, \neg R, S\}$	given	7.	$\{\neg S, \neg T\}$	$\mathbf{3}, 4$
$\mathbf{3}$.	$\{R\}$	given	$\mathbf{8}$.	$\{S\}$	$\mathbf{1}, 6$
4.	$\{\neg R, \neg S, \neg T\}$	given	9.	$\{\neg R, \neg T\}$	$4, \mathbf{8}$
5.	$\{\neg R, S\}$	$\mathbf{1}, 2$	$\mathbf{10}$.	$\{\neg T\}$	$7, \mathbf{8}$

(The unit clauses have boldface numbers.)

No further clauses can be derived by unit resolution.

Thus the empty clause cannot be derived, so $\mathcal{S}$ is satisfiable.

A diagram of the unit resolution steps in the previous example:



Graph Clauses

Let $\mathbf{G}$ be a finite graph with vertex set V and edge set E.

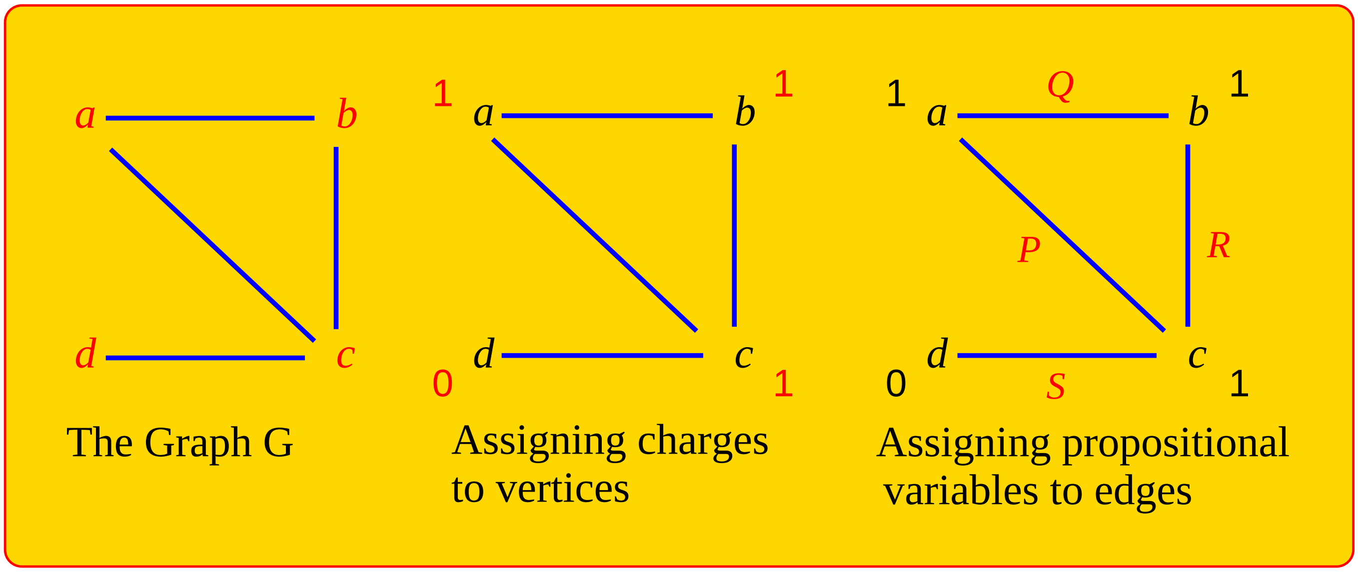
Label each vertex v with 0 or 1.

This number, $charge(v)$, is the **charge** of the vertex.

The **total charge** of the graph is the sum of the vertex charges (modulo 2).

Label the edges with distinct propositional variables.

For v a vertex, the set of variables labelling edges adjacent to v is $Var(v)$.



For $v \in V$ construct $Clauses(v)$, as follows:

C is in $Clauses(v)$ iff

- the propositional variables in C are precisely those of $Var(v)$

- the number of negative literals in C is not equal to $charge(v)$ (modulo 2).

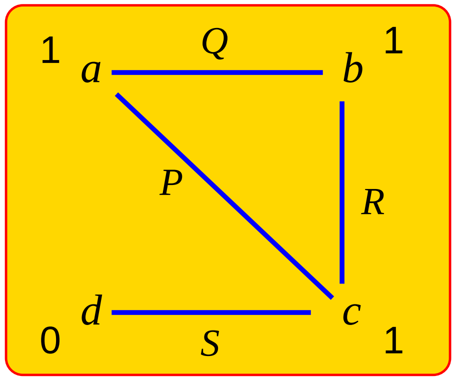
Let $\widehat{\mathbf{G}}$ be the labelled graph.

Then $Clauses(\widehat{\mathbf{G}})$ is the union of the various $Clauses(v)$.

Theorem [Tseitin, 1968]
$Clauses(\widehat{\mathbf{G}})$ is satisfiable iff the total charge is zero.

Thus we have a very fast test to see if $Clauses(\widehat{\mathbf{G}})$ is satisfiable, but Tseitin showed that to decide this using resolution can be very slow.

For the example



we have

$$Clauses(a) \;=\; \{\{P, Q\},\ \{\neg P, \neg Q\}\}$$

$$Clauses(b) \;=\; \{\{Q, R\},\ \{\neg Q, \neg R\}$$

$$Clauses(c) \;=\; \{\{P, R, S\},\ \{\neg P, R, \neg S\},\ \{\neg P, \neg R, S\},\ \{P, \neg R, \neg S\}\}$$

$$Clauses(d) \;=\; \{\neg S\}.$$

Because the total charge is 1, by Tseitin's theorem this set of 9 clauses is not satisfiable.

Pigeonhole Clauses

In 1974 Cook and Rechkow suggested that the set of clauses expressing a pigeonhole principle would be difficult to prove unsatisfiable by resolution.

The **pigeonhole principle** $\mathbf{P_n}$: one cannot put $n+1$ objects into n slots with distinct objects going into distinct slots.

We choose propositional variables P_{ij} for $1 \le i \le n+1$ and $1 \le j \le n$.

Our intended interpretation of P_{ij} is that the ith object goes into the jth slot.

So we write down the following clauses:

- $\{P_{i1}, \cdots, P_{in}\}$ for $1 \le i \le n+1$.

These say that each object i goes into some slot k .

- $\{\neg P_{ik}, \neg P_{jk}\}$ for $1 \le i < j \le n+1$, $1 \le k \le n$.

These say that distinct objects i and j cannot go into the same slot.

Of course, this cannot be done, so the clauses must be unsatisfiable.

However, if we throw away any one clause from $\mathbf{P_n}$, the remaining collection of clauses is satisfiable!

Example 2.13.1

$\mathbf{P_2}$ is the set of nine clauses in six variables:

$$
\begin{array}{lll}
\{P_{11},\ P_{12}\} & \{P_{21},\ P_{22}\} & \{P_{31},\ P_{32}\} \\
\{\neg\ P_{11},\ \neg\ P_{21}\} & \{\neg\ P_{12},\ \neg\ P_{22}\} & \{\neg\ P_{11},\ \neg\ P_{31}\} \\
\{\neg\ P_{12},\ \neg\ P_{32}\} & \{\neg\ P_{21},\ \neg\ P_{31}\} & \{\neg\ P_{22},\ \neg\ P_{32}\}.
\end{array}
$$

Note: In 1985 A. Haken proved that pigeonhole clauses are indeed difficult for resolution.