

Propositional Logic

Basics

Syntax of propositional logic

Definition

An **atomic formula** (or **atom**) has the form A_i where $i = 1, 2, 3, \dots$

Formulas are defined inductively:

- ▶ \perp (“False”) and \top (“True”) are formulas
- ▶ All atomic formulas are formulas
- ▶ For all formulas F , $\neg F$ is a formula.
- ▶ For all formulas F and G , $(F \circ G)$ is a formula, where $\circ \in \{\wedge, \vee, \rightarrow\}$

\neg	is called	negation
\wedge	is called	conjunction
\vee	is called	disjunction
\rightarrow	is called	implication

Parentheses

Precedence of logical operators in decreasing order:

$$\neg \quad \wedge \quad \vee \quad \rightarrow$$

Operators with higher precedence bind more strongly.

Example

Instead of $(A \rightarrow ((B \wedge \neg(C \vee D)) \vee E))$

we can write $A \rightarrow B \wedge \neg(C \vee D) \vee E$.

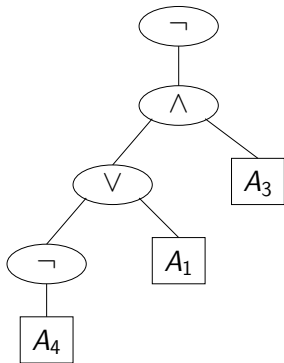
Outermost parentheses can be dropped.

Syntax tree of a formula

Every formula can be represented by a syntax tree.

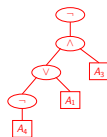
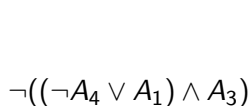
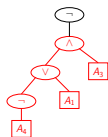
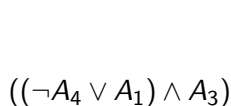
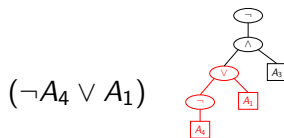
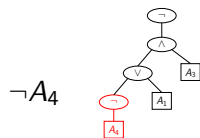
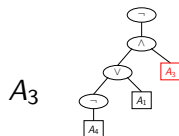
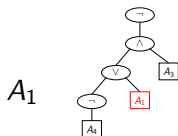
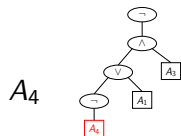
Example

$$F = \neg((\neg A_4 \vee A_1) \wedge A_3)$$



Subformulas

The **subformulas** of a formula are the formulas corresponding to the subtrees of its syntax tree.



Induction on formulas

Proof by induction on the structure of a formula:

In order to prove some property $\mathcal{P}(F)$ for all formulas F it suffices to prove the following:

- ▶ Base cases:
prove $\mathcal{P}(\perp)$, prove $\mathcal{P}(\top)$, and prove $\mathcal{P}(A_i)$ for all atoms A_i
- ▶ Induction step for \neg :
prove $\mathcal{P}(\neg F)$ under the induction hypothesis $\mathcal{P}(F)$
- ▶ Induction step for all $\circ \in \{\wedge, \vee, \rightarrow\}$:
prove $\mathcal{P}(F \circ G)$ under the induction hypotheses $\mathcal{P}(F)$ and $\mathcal{P}(G)$

Operators that are merely abbreviations need not be considered!

Semantics of propositional logic (I)

The elements of the set $\{0, 1\}$ are called **truth values**.
(You may call 0 “false” and 1 “true”)

An **assignment** is a function $\mathcal{A} : Atoms \rightarrow \{0, 1\}$
where *Atoms* is the set of all atoms.

We extend \mathcal{A} to a function $\hat{\mathcal{A}} : Formulas \rightarrow \{0, 1\}$

Semantics of propositional logic (II)

$$\hat{\mathcal{A}}(A_i) = \mathcal{A}(A_i)$$

$$\hat{\mathcal{A}}(\neg F) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\mathcal{A}}(F \wedge G) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 1 \text{ and } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\mathcal{A}}(F \vee G) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 1 \text{ or } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\mathcal{A}}(F \rightarrow G) = \begin{cases} 1 & \text{if } \hat{\mathcal{A}}(F) = 0 \text{ or } \hat{\mathcal{A}}(G) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Instead of $\hat{\mathcal{A}}$ we simply write \mathcal{A}

Using arithmetic: $\mathcal{A}(F \wedge G) = \min(\mathcal{A}(F), \mathcal{A}(G))$

$\mathcal{A}(F \vee G) = \max(\mathcal{A}(F), \mathcal{A}(G))$

Abbreviations

$A, B, C,$
 $P, Q, R,$ or ... instead of $A_1, A_2, A_3 \dots$

$$F_1 \leftrightarrow F_2 \text{ abbreviates } (F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2)$$
$$\bigvee_{i=1}^n F_i \text{ abbreviates } (\dots ((F_1 \vee F_2) \vee F_3) \vee \dots \vee F_n)$$
$$\bigwedge_{i=1}^n F_i \text{ abbreviates } (\dots ((F_1 \wedge F_2) \wedge F_3) \wedge \dots \wedge F_n)$$

Special cases:

$$\bigvee_{i=1}^0 F_i = \bigvee \emptyset = \perp \qquad \bigwedge_{i=1}^0 F_i = \bigwedge \emptyset = \top$$

Truth tables (I)

We can compute \hat{A} with the help of **truth tables**.

\neg	A	A	\vee	B	A	\wedge	B
1	0	0	0	0	0	0	0
0	1	0	1	1	0	0	1
		1	1	0	1	0	0
		1	1	1	1	1	1

A	\rightarrow	B	A	\leftrightarrow	B
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	1	1	1	1	1

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r) \leftrightarrow \neg r$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r) \leftrightarrow \neg r$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r)$	\leftrightarrow	\neg	r
0	0	0	0			0
0	0	1	0			1
0	1	0	0			0
0	1	1	0			1
1	0	0	1			0
1	0	1	1			1
1	1	0	1			0
1	1	1	1			1

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r)$	\leftrightarrow	\neg	r
0	0	0	0	1	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	0	1	1	1	0	1
1	1	0	1	0	1	0
1	1	1	1	0	1	1

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r)$	\leftrightarrow	\neg	r
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	0	0	1
0	1	1	0	0	0	1
1	0	0	1	1	1	0
1	0	1	1	1	1	0
1	1	0	1	1	0	1
1	1	1	1	1	0	1

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r)$	\leftrightarrow	\neg	r
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	0	0	1
0	1	1	0	0	0	1
1	0	0	1	1	1	0
1	0	1	1	1	1	0
1	1	0	1	1	0	1
1	1	1	1	1	0	1

Truth table of a formula

p	q	r	$((p \vee \neg q) \rightarrow r)$	\leftrightarrow	\neg	r
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	0	0	1
0	1	1	0	0	0	1
1	0	0	1	1	1	0
1	0	1	1	1	1	0
1	1	0	1	1	0	0
1	1	1	1	1	0	1

Coincidence Lemma

Lemma

Let \mathcal{A}_1 and \mathcal{A}_2 be two assignments.

*If $\mathcal{A}_1(A_i) = \mathcal{A}_2(A_i)$ for all atoms A_i in some formula F ,
then $\mathcal{A}_1(F) = \mathcal{A}_2(F)$.*

Proof.

Exercise.



Models

If $\mathcal{A}(F) = 1$ then we write $\mathcal{A} \models F$
and say F is true under \mathcal{A}
or \mathcal{A} is a model of F

If $\mathcal{A}(F) = 0$ then we write $\mathcal{A} \not\models F$
and say F is false under \mathcal{A}
or \mathcal{A} is not a model of F

Validity and satisfiability

Definition (Validity)

A formula F is **valid** (or a **tautology**) if every assignment is a model of F .

We write $\models F$ if F is valid, and $\not\models F$ otherwise.

Definition (Satisfiability)

A formula F is **satisfiable** if it has at least one model; otherwise F is **unsatisfiable**.

A (finite or infinite!) set of formulas S is **satisfiable** if there is an assignment that is a model of every formula in S .

Exercise

	Valid	Satisfiable	Unsatisfiable
A			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x		

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x		

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x	x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x	x	
$A \rightarrow (A \rightarrow B)$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x	x	
$A \rightarrow (A \rightarrow B)$		x	

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x	x	
$A \rightarrow (A \rightarrow B)$		x	
$A \leftrightarrow \neg A$			

Exercise

	Valid	Satisfiable	Unsatisfiable
A		x	
$A \vee B$		x	
$A \vee \neg A$	x	x	
$A \wedge \neg A$			x
$A \rightarrow \neg A$		x	
$A \rightarrow (B \rightarrow A)$	x	x	
$A \rightarrow (A \rightarrow B)$		x	
$A \leftrightarrow \neg A$			x

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable		

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		T

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		T
If F is valid then $\neg F$ is unsatisfiable		

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		T
If F is valid then $\neg F$ is unsatisfiable	Y	

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		T
If F is valid then $\neg F$ is unsatisfiable	Y	
If F is unsatisfiable then $\neg F$ is unsatisfiable		

Exercise

Which of the following statements are true?

	Y	C.ex.
If F is valid then F is satisfiable	Y	
If F is satisfiable then $\neg F$ is satisfiable		T
If F is valid then $\neg F$ is unsatisfiable	Y	
If F is unsatisfiable then $\neg F$ is unsatisfiable		\perp

Mirroring principle

all propositional formulas

valid formulas G	satisfiable but not valid formulas F $\neg F$	unsatisfiable formulas $\neg G$
------------------------------	--	---

Consequence (aka entailment)

Definition

A formula G is a (semantic) consequence of a set of formulas M if every assignment that is a model of every $F \in M$ is also a model of G .

We also say that M entails G and write $M \models G$.

In a nutshell:

“Every model of M is a model of G .”

Example

$A \vee B, A \rightarrow B, B \wedge R \rightarrow \neg A, R \models (R \wedge \neg A) \wedge B$

Consequence

Example

$$\underbrace{A \vee B, A \rightarrow B, B \wedge R \rightarrow \neg A, R}_M \models (R \wedge \neg A) \wedge B$$

Proof:

Assume $\mathcal{A} \models F$ for all $F \in M$.

We need to prove $\mathcal{A} \models (R \wedge \neg A) \wedge B$.

It suffices to prove $\mathcal{A} \models R$, $\mathcal{A} \models \neg A$, and $\mathcal{A} \models B$

- ▶ $\mathcal{A} \models R$ is immediate.
- ▶ $\mathcal{A} \models B$ follows from $\mathcal{A} \models A \vee B$ and $\mathcal{A} \models A \rightarrow B$:

Proof by cases:

If $\mathcal{A}(A) = 0$ then $\mathcal{A}(B) = 1$ because $\mathcal{A} \models A \vee B$

If $\mathcal{A}(A) = 1$ then $\mathcal{A}(B) = 1$ because $\mathcal{A} \models A \rightarrow B$

- ▶ $\mathcal{A} \models \neg A$ follows from $\mathcal{A} \models B$ and $\mathcal{A} \models R$.

Exercise

M	F	$M \models F ?$
A	$A \vee B$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$	$(A \rightarrow B) \rightarrow C$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$	$(A \rightarrow B) \rightarrow C$	N

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$	$(A \rightarrow B) \rightarrow C$	N
$(A \rightarrow B) \rightarrow C$		

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$	$(A \rightarrow B) \rightarrow C$	N
$(A \rightarrow B) \rightarrow C$	$A \rightarrow (B \rightarrow C)$	

Exercise

M	F	$M \models F ?$
A	$A \vee B$	Y
A	$A \wedge B$	N
A, B	$A \vee B$	Y
A, B	$A \wedge B$	Y
$A \vee B$	A	N
$A, A \rightarrow B$	B	Y
$A \rightarrow (B \rightarrow C)$	$(A \rightarrow B) \rightarrow C$	N
$(A \rightarrow B) \rightarrow C$	$A \rightarrow (B \rightarrow C)$	Y

Consequence

Exercise

The following statements are equivalent:

1. $F_1, \dots, F_k \models G$
2. $\models (\bigwedge_{i=1}^k F_i) \rightarrow G$

Proof of “if $F_1, \dots, F_k \models G$ then $\models \underbrace{(\bigwedge_{i=1}^k F_i) \rightarrow G}_H$ ”.

Assume $F_1, \dots, F_k \models G$.

We need to prove $\models H$, i.e. $\mathcal{A}(H) = 1$ for all \mathcal{A} .

We pick an arbitrary \mathcal{A} and show $\mathcal{A}(H) = 1$.

Proof by cases: either $\mathcal{A}(\bigwedge F_i) = 0$ or $\mathcal{A}(\bigwedge F_i) = 1$.

- ▶ $\mathcal{A}(\bigwedge F_i) = 0$: Then $\mathcal{A}(H) = 1$ because $H = \bigwedge F_i \rightarrow G$.
- ▶ $\mathcal{A}(\bigwedge F_i) = 1$: Then $\mathcal{A}(F_i) = 1$ for all i .
Therefore \mathcal{A} is a model of F_1, \dots, F_k .
Therefore $\mathcal{A} \models G$ because $F_1, \dots, F_k \models G$.

Validity and satisfiability

Exercise

The following statements are equivalent:

1. $F \rightarrow G$ is valid.
2. $F \wedge \neg G$ is unsatisfiable.

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.		

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.		

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	
If $F \in M$ then $M \models F$.		

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	
If $F \in M$ then $M \models F$.	Y	

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	
If $F \in M$ then $M \models F$.	Y	
If $F \models G$ then $\neg F \models \neg G$.		

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	
If $F \in M$ then $M \models F$.	Y	
If $F \models G$ then $\neg F \models \neg G$.	N	

Exercise

Let M be a set of formulas, and let F and G be formulas.
Which of the following statements hold?

	Y/N	C.ex.
If F satisfiable then $M \models F$.	N	$\neg A \models A$
If F valid then $M \models F$.	Y	
If $F \in M$ then $M \models F$.	Y	
If $F \models G$ then $\neg F \models \neg G$.	N	$A \models A \vee B$

Notation

Warning: The symbol \models is overloaded:

$$\mathcal{A} \models F$$

$$\models F$$

$$M \models F$$

Convenient variations for set of formulas S :

$\mathcal{A} \models S$ means that for all $F \in S$, $\mathcal{A} \models F$

$\models S$ means that for all $F \in S$, $\models F$

$M \models S$ means that for all $F \in S$, $M \models F$

Propositional Logic Equivalences

Equivalence

Definition (Equivalence)

Two formulas F and G are (semantically) equivalent if $\mathcal{A}(F) = \mathcal{A}(G)$ for every assignment \mathcal{A} .

We write $F \equiv G$ to denote that F and G are equivalent.

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Exercise

Which of the following equivalences hold?

$$(A \wedge (A \vee B)) \equiv A$$

$$(A \wedge (B \vee C)) \equiv ((A \wedge B) \vee C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \rightarrow B) \rightarrow C)$$

$$(A \rightarrow (B \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

$$(A \rightarrow B) \equiv (\neg A \rightarrow \neg B)$$

$$(A \leftrightarrow (B \leftrightarrow C)) \equiv ((A \leftrightarrow B) \leftrightarrow C)$$

Observation

The following connections hold:

$$\begin{aligned} \models F \rightarrow G & \text{ iff } F \models G \\ \models F \leftrightarrow G & \text{ iff } F \equiv G \end{aligned}$$

NB: “iff” means “if and only if”

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:
- ▶ **Unsatisfiability** to **Validity**:
- ▶ **Validity** to **Consequence**:
- ▶ **Consequence** to **Validity**:
- ▶ **Validity** to **Equivalence**:
- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

F valid iff ? unsatisfiable

- ▶ **Unsatisfiability** to **Validity**:

- ▶ **Validity** to **Consequence**:

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

- ▶ **Validity** to **Consequence**:

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad ? \text{ valid}$$

- ▶ **Validity** to **Consequence**:

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

F valid iff $\neg F$ unsatisfiable

- ▶ **Unsatisfiability** to **Validity**:

F unsatisfiable iff $\neg F$ valid

- ▶ **Validity** to **Consequence**:

F valid iff $? \models ?$

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad ? \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad F \rightarrow G \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad F \rightarrow G \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

$$F \text{ valid} \quad \text{iff} \quad ? \equiv ?$$

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad F \rightarrow G \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

$$F \text{ valid} \quad \text{iff} \quad F \equiv \top$$

- ▶ **Equivalence** to **Validity**:

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad F \rightarrow G \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

$$F \text{ valid} \quad \text{iff} \quad F \equiv \top$$

- ▶ **Equivalence** to **Validity**:

$$F \equiv G \quad \text{iff} \quad ? \text{ valid}$$

Reductions between problems (I)

- ▶ **Validity** to **Unsatisfiability**:

$$F \text{ valid} \quad \text{iff} \quad \neg F \text{ unsatisfiable}$$

- ▶ **Unsatisfiability** to **Validity**:

$$F \text{ unsatisfiable} \quad \text{iff} \quad \neg F \text{ valid}$$

- ▶ **Validity** to **Consequence**:

$$F \text{ valid} \quad \text{iff} \quad \top \models F$$

- ▶ **Consequence** to **Validity**:

$$F \models G \quad \text{iff} \quad F \rightarrow G \text{ valid}$$

- ▶ **Validity** to **Equivalence**:

$$F \text{ valid} \quad \text{iff} \quad F \equiv \top$$

- ▶ **Equivalence** to **Validity**:

$$F \equiv G \quad \text{iff} \quad F \leftrightarrow G \text{ valid}$$

Properties of semantic equivalence

- ▶ Semantic equivalence is an **equivalence relation** between formulas.
- ▶ Semantic equivalence is **closed under operators**:

If $F_1 \equiv F_2$ and $G_1 \equiv G_2$

then $\neg F_1 \equiv \neg F_2$ and

$(F_1 \circ G_1) \equiv (F_2 \circ G_2)$ for $\circ \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$

Equivalence relation + Closure under Operations

=

Congruence relation

Replacement theorem

Theorem

Let $F \equiv G$. Let H be a formula with an occurrence of F as a subformula. Let H' be the result of replacing an arbitrary occurrence of F in H by G . Then $H \equiv H'$.

Proof by induction on the structure of H .

We consider only the case $H = \neg H_0$.

Two cases: either $F = H$ or F is a subformula of H_0 .

- ▶ $F = H$: Then $H' = G$ and thus $H = F \equiv G = H'$.
- ▶ F is a subformula of H_0 .

Let H'_0 be the result of replacing F by G in H_0 .

IH: $H_0 \equiv H'_0$

Thus $H = \neg H_0 \equiv \neg H'_0 = H'$.

Equivalences (I)

Theorem

$$(F \wedge F) \equiv F$$

$$(F \vee F) \equiv F$$

$$(F \wedge G) \equiv (G \wedge F)$$

$$(F \vee G) \equiv (G \vee F)$$

$$((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H))$$

$$((F \vee G) \vee H) \equiv (F \vee (G \vee H))$$

$$(F \wedge (F \vee G)) \equiv F$$

$$(F \vee (F \wedge G)) \equiv F$$

(Idempotence)

(Commutativity)

(Associativity)

(Absorption)

Equivalences (II)

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$$

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$$

(Distributivity)

$$\neg\neg F \equiv F$$

(Double negation)

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$$

$$\neg(F \vee G) \equiv (\neg F \wedge \neg G)$$

(deMorgan's Laws)

$$\neg\top \equiv \perp$$

$$\neg\perp \equiv \top$$

$$(\top \vee G) \equiv \top$$

$$(\top \wedge G) \equiv G$$

$$(\perp \vee G) \equiv G$$

$$(\perp \wedge G) \equiv \perp$$

Warning

The symbols \models and \equiv are **not** operators
in the language of propositional logic
but part of the meta-language for talking about logic.

Examples:

$\mathcal{A} \models F$ and $F \equiv G$ are not propositional formulas.
 $(\mathcal{A} \models F) \equiv G$ and $(F \equiv G) \leftrightarrow (G \equiv F)$ are nonsense.

Propositional Logic

Normal Forms

Abbreviations

Until further notice, we consider the reduced syntax without \top , \perp and only \neg , \vee , \wedge as operators:

$F_1 \rightarrow F_2$ abbreviates $\neg F_1 \vee F_2$

$F_1 \leftrightarrow F_2$ abbreviates $(F_1 \wedge F_2) \vee (\neg F_1 \wedge \neg F_2)$

\top abbreviates $A_1 \vee \neg A_1$

\perp abbreviates $A_1 \wedge \neg A_1$

Literals

Definition

A **literal** is an atom or the negation of an atom.
In the former case the literal is **positive**,
in the latter case it is **negative**.

Negation Normal Form (NNF)

Definition

A formula is in **negation normal form (NNF)** if negation (\neg) occurs only directly in front of atoms.

Example

In NNF: $\neg A \wedge \neg B$

Not in NNF: $\neg(A \vee B)$

Transformation into NNF

Any formula can be transformed into an equivalent formula in NNF by pushing \neg inwards. Apply the following equivalences from left to right as long as possible:

$$\begin{aligned}\neg\neg F &\equiv F \\ \neg(F \wedge G) &\equiv (\neg F \vee \neg G) \\ \neg(F \vee G) &\equiv (\neg F \wedge \neg G)\end{aligned}$$

Example

$$\begin{aligned}(\neg(A \wedge \neg B) \wedge C) &\equiv ((\neg A \vee \neg\neg B) \wedge C) \equiv ((\neg A \vee B) \wedge C) \\ (\text{"}F \equiv G \equiv H\text{" is an abbreviation for "}F \equiv G \text{ and } G \equiv H\text{"})\end{aligned}$$

Does this process always terminate? Is the result unique?

Transformation into NNF

Any formula can be transformed into an equivalent formula in NNF by pushing \neg inwards. Apply the following equivalences from left to right as long as possible:

$$\begin{aligned}\neg\neg F &\equiv F \\ \neg(F \wedge G) &\equiv (\neg F \vee \neg G) \\ \neg(F \vee G) &\equiv (\neg F \wedge \neg G)\end{aligned}$$

Example

$$\begin{aligned}(\neg(A \wedge \neg B) \wedge C) &\equiv ((\neg A \vee \neg\neg B) \wedge C) \equiv ((\neg A \vee B) \wedge C) \\ (\text{"}F \equiv G \equiv H\text{" is an abbreviation for "}F \equiv G \text{ and } G \equiv H\text{"})\end{aligned}$$

Does this process always terminate? Is the result unique?

CNF and DNF

Definition

A formula F is in **conjunctive normal form (CNF)** if it is a conjunction of disjunctions of literals:

$$F = \left(\bigwedge_{i=1}^n \left(\bigvee_{j=1}^{m_i} L_{i,j} \right) \right),$$

where $L_{i,j} \in \{A_1, A_2, \dots\} \cup \{\neg A_1, \neg A_2, \dots\}$

CNF and DNF

Definition

A formula F is in **conjunctive normal form (CNF)** if it is a conjunction of disjunctions of literals:

$$F = \left(\bigwedge_{i=1}^n \left(\bigvee_{j=1}^{m_i} L_{i,j} \right) \right),$$

where $L_{i,j} \in \{A_1, A_2, \dots\} \cup \{\neg A_1, \neg A_2, \dots\}$

Definition

A formula F is in **disjunctive normal form (DNF)** if it is a disjunction of conjunctions of literals:

$$F = \left(\bigvee_{i=1}^n \left(\bigwedge_{j=1}^{m_i} L_{i,j} \right) \right),$$

where $L_{i,j} \in \{A_1, A_2, \dots\} \cup \{\neg A_1, \neg A_2, \dots\}$

Transformation into CNF and DNF

Any formula can be transformed into an equivalent formula in CNF or DNF in two steps:

1. Transform the initial formula into its NNF
2. Transform the NNF into CNF or DNF:
 - ▶ Transformation into CNF. Apply the following equivalences from left to right as long as possible:

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$$

$$((F \wedge G) \vee H) \equiv ((F \vee H) \wedge (G \vee H))$$

- ▶ Transformation into DNF. Apply the following equivalences from left to right as long as possible:

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$$

$$((F \vee G) \wedge H) \equiv ((F \wedge H) \vee (G \wedge H))$$

Transformation into CNF and DNF

Any formula can be transformed into an equivalent formula in CNF or DNF in two steps:

1. Transform the initial formula into its NNF
2. Transform the NNF into CNF or DNF:
 - ▶ Transformation into CNF. Apply the following equivalences from left to right as long as possible:

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$$

$$((F \wedge G) \vee H) \equiv ((F \vee H) \wedge (G \vee H))$$

- ▶ Transformation into DNF. Apply the following equivalences from left to right as long as possible:

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$$

$$((F \vee G) \wedge H) \equiv ((F \wedge H) \vee (G \wedge H))$$

Termination

Why does the transformation into NNF and CNF terminate?

Challenge Question: Find a weight function $w :: \text{formula} \rightarrow \mathbb{N}$ such that $w(l.h.s.) > w(r.h.s.)$ for the equivalences

$$\begin{aligned}\neg\neg F &\equiv F \\ \neg(F \wedge G) &\equiv (\neg F \vee \neg G) \\ \neg(F \vee G) &\equiv (\neg F \wedge \neg G) \\ (F \vee (G \wedge H)) &\equiv ((F \vee G) \wedge (F \vee H)) \\ ((F \wedge G) \vee H) &\equiv ((F \vee H) \wedge (G \vee H))\end{aligned}$$

Define w recursively:

$$w(A_i) = \dots$$

$$w(\neg F) = \dots w(F) \dots$$

$$w(F \wedge G) = \dots w(F) \dots w(G) \dots$$

$$w(F \vee G) = \dots w(F) \dots w(G) \dots$$

Complexity considerations

The CNF and DNF of a formula of size n can have size 2^n

Can we do better? Yes, if we do not insist on \equiv .

Definition

Two formulas F and G are **equisatisfiable** if F is satisfiable iff G is satisfiable.

Theorem

For every formula F of size n there is an equisatisfiable CNF formula G of size $O(n)$.

Propositional Logic
Definitional CNF
(Tseytin's transformation)

Definitional CNF

1. The **definitional CNF** of a formula is obtained in 2 steps:

Repeatedly replace a subformula G of the form $\neg A$, $A \wedge B$ or $A \vee B$ (A, B atoms!) by a new atom A' and conjoin $A' \leftrightarrow G$. (This replacement is not applied to the “definitions” $A' \leftrightarrow G$ but only to the (remains of the) original formula.)

2. Translate all the subformulas $A' \leftrightarrow G$ into CNF.

Example

$$\begin{aligned} & \neg(\boxed{A_1 \vee A_2}) \wedge A_3 \\ \rightsquigarrow & \boxed{\neg A_4} \wedge A_3 \wedge (A_4 \leftrightarrow (A_1 \vee A_2)) \\ \rightsquigarrow & \boxed{A_5 \wedge A_3} \wedge (A_4 \leftrightarrow (A_1 \vee A_2)) \wedge (A_5 \leftrightarrow \neg A_4) \\ \rightsquigarrow & A_6 \wedge (A_4 \leftrightarrow (A_1 \vee A_2)) \wedge (A_5 \leftrightarrow \neg A_4) \wedge (A_6 \leftrightarrow (A_5 \wedge A_3)) \\ \rightsquigarrow & A_6 \wedge \text{CNF}(A_4 \leftrightarrow (A_1 \vee A_2)) \wedge \text{CNF}(A_5 \leftrightarrow \neg A_4) \wedge \text{CNF}(A_6 \leftrightarrow (A_5 \wedge A_3)) \end{aligned}$$

Definitional CNF: Complexity

Let the initial formula have size n .

1. Each replacement step increases the size of the formula by a constant.
There are at most as many replacement steps as subformulas, linearly many.
2. The conversion of each $A \leftrightarrow G$ into CNF increases the size by a constant.
There are only linearly many such subformulas.

Thus: the definitional CNF has size $O(n)$, and can be constructed in $O(n)$ time.

Definitional CNF: Correctness — Notation

Definition

The notation $F[G/A]$ denotes the result of replacing **all** occurrences of the atom A in F by G .

We pronounce it as “ F with G for A ”.

Example

$$(A \wedge B)[(A \rightarrow B)/B] = (A \wedge (A \rightarrow B))$$

Definition

The notation $\mathcal{A}[v/A]$ denotes a modified version of \mathcal{A} that maps A to v and behaves like \mathcal{A} otherwise:

$$(\mathcal{A}[v/A])(A_i) = \begin{cases} v & \text{if } A_i = A \\ \mathcal{A}(A_i) & \text{otherwise} \end{cases}$$

Definitional CNF: Correctness — Substitution Lemma

Lemma

$\mathcal{A}(F[G/A]) = \mathcal{A}'(F)$ where $\mathcal{A}' = \mathcal{A}[\mathcal{A}(G)/A]$

Proof by structural induction on F .

► F is an atom:

If $F = A$: $\mathcal{A}(F[G/A]) = \mathcal{A}(G) = \mathcal{A}'(F)$

If $F \neq A$: $\mathcal{A}(F[G/A]) = \mathcal{A}(F) = \mathcal{A}'(F)$

► $F = F_1 \wedge F_2$:

$$\begin{aligned}\mathcal{A}((F_1 \wedge F_2)[G/A]) &= \mathcal{A}(F_1[G/A] \wedge F_2[G/A]) \\ &= \min(\mathcal{A}(F_1[G/A]), \mathcal{A}(F_2[G/A])) \\ &\stackrel{IH}{=} \min(\mathcal{A}'(F_1), \mathcal{A}'(F_2)) \\ &= \mathcal{A}'(F_1 \wedge F_2)\end{aligned}$$

Definitional CNF: Correctness

Each replacement step produces an equisatisfiable formula:

Lemma

Let A be an atom that does not occur in G .

Then $F[G/A]$ is equisatisfiable with $F \wedge (A \leftrightarrow G)$.

Proof Assume $\mathcal{A} \models F[G/A]$ for some assignment \mathcal{A} .

Let $\mathcal{A}' := \mathcal{A}[\mathcal{A}(G)/A]$. We prove $\mathcal{A}' \models F \wedge (A \leftrightarrow G)$.

$\mathcal{A}' \models F$: Substitution Lemma.

$\mathcal{A}' \models (A \leftrightarrow G)$: Because $\mathcal{A}'(A) = \mathcal{A}(G) = \mathcal{A}'(G)$
(by definition of \mathcal{A}' and because A does not occur in G).

Assume $\mathcal{A} \models F \wedge (A \leftrightarrow G)$ for some assignment \mathcal{A} .

We prove $\mathcal{A} \models F[G/A]$, that is, $\mathcal{A}(F[G/A]) = 1$.

We show $\mathcal{A}(F[G/A]) = \mathcal{A}'(F) = \mathcal{A}(F) = 1$ for $\mathcal{A}' := \mathcal{A}[\mathcal{A}(G)/A]$.

$\mathcal{A}(F[G/A]) = \mathcal{A}'(F)$: Substitution Lemma.

$\mathcal{A}'(F) = \mathcal{A}(F)$: From $\mathcal{A} \models (A \leftrightarrow G)$ follows $\mathcal{A}(A) = \mathcal{A}(G)$, and
so $\mathcal{A}' = \mathcal{A}$.

$\mathcal{A}(F) = 1$: Because $\mathcal{A} \models F$.

Definitional CNF: Correctness

Does $F \wedge (A \leftrightarrow G) \models F[G/A]$ hold?

Does $F[G/A] \models F \wedge (A \leftrightarrow G)$ hold?

Summary

Theorem

For every formula F of size n
there is an *equisatisfiable CNF* formula G of size $O(n)$.

Proof.

Repeated application of the Lemma. □

Similarly it can be shown:

Theorem

For every formula F of size n
there is an *equivalent DNF* formula G of size $O(n)$.

Validity of CNF

Validity of formulas in CNF can be checked in linear time.

A formula in CNF is valid iff all its disjunctions are valid.

A disjunction is valid iff it contains both an atomic A and $\neg A$ as literals.

Example

Valid: $(A \vee \neg A \vee B) \wedge (C \vee \neg C)$

Not valid: $(A \vee \neg A) \wedge (\neg A \vee C)$

Satisfiability of DNF

Satisfiability of formulas in DNF can be checked in linear time.

A formula in DNF is satisfiable iff at least one of its conjunctions is satisfiable. A conjunction is satisfiable iff it does not contain both an atomic A and $\neg A$ as literals.

Example

Satisfiable: $(\neg B \wedge A \wedge B) \vee (\neg A \wedge C)$

Unsatisfiable: $(A \wedge \neg A \wedge B) \vee (C \wedge \neg C)$

Satisfiability/validity of DNF and CNF

Theorem

Satisfiability of formulas in CNF is NP-complete.

Theorem

Validity of formulas in DNF is co-NP-complete.

Standard decision procedure for validity of F :

1. Transform $\neg F$ into an equisat. formula G in def. CNF
2. Apply efficient CNF-based SAT solver to G

Propositional Logic

Horn Formulas

Efficient satisfiability checks

In this and the next slide sets:

- ▶ A very efficient satisfiability check for a special class of formulas in CNF: **Horn formulas**,
- ▶ Efficient satisfiability checks for arbitrary formulas in CNF: **DPLL** and **resolution** (later).

Horn formulas

Definition

A formula F in CNF is a **Horn formula** if every disjunction in F contains at most one positive literal.

Every disjunct of a Horn formula can equivalently be viewed as an implication $K \rightarrow B$ where

- ▶ K is a conjunction of atoms or \top , and
- ▶ B is an atom or \perp .

$$\begin{aligned} A &\equiv (\top \rightarrow A) \\ (\neg A \vee \neg B \vee C) &\equiv (A \wedge B \rightarrow C) \\ (\neg A \vee B) &\equiv (A \rightarrow B) \\ \neg A &\equiv (A \rightarrow \perp) \\ (\neg A \vee \neg B) &\equiv (A \wedge B \rightarrow \perp) \end{aligned}$$

Horn formulas

Definition

A formula F in CNF is a **Horn formula** if every disjunction in F contains at most one positive literal.

Every disjunct of a Horn formula can equivalently be viewed as an implication $K \rightarrow B$ where

- ▶ K is a conjunction of atoms or \top , and
- ▶ B is an atom or \perp .

A	\equiv	$(\top \rightarrow A)$	fact
$(\neg A \vee \neg B \vee C)$	\equiv	$(A \wedge B \rightarrow C)$	rule
$(\neg A \vee B)$	\equiv	$(A \rightarrow B)$	rule
$\neg A$	\equiv	$(A \rightarrow \perp)$	goal
$(\neg A \vee \neg B)$	\equiv	$(A \wedge B \rightarrow \perp)$	goal

Satisfiability check for Horn formulas

Horn: $(\neg A \vee \neg B \vee C) \equiv (A \wedge B \rightarrow C)$

Non-Horn: $(\neg A \vee \neg B \vee C \vee D) \equiv (A \wedge B \rightarrow C \vee D)$

Satisfiability check for Horn formulas

Input: a Horn formula F .

Output: Model \mathcal{M} of F or “unsatisfiable”

```
for all atoms  $A_i$  in  $F$  do  $\mathcal{M}(A_i) := 0$ ;  
while  $F$  has a conjunct  $K \rightarrow B$   
    such that  $\mathcal{M}(K) = 1$  and  $\mathcal{M}(B) = 0$   
do  
    if  $B = \perp$  then return “unsatisfiable”  
    else  $\mathcal{M}(B) := 1$   
return  $\mathcal{M}$ 
```

Maximal number of iterations of the while loop:
number of implications in F

Each iteration requires at most $O(|F|)$ steps.

Overall complexity: $O(|F|^2)$

[Algorithm can be improved to $O(|F|)$. See Schöning.]

Correctness of the model building algorithm

Theorem

The algorithm returns a model iff F is satisfiable.

Proof. Invariant: if $\mathcal{M}(A) = 1$, then $\mathcal{A}(A) = 1$ for every atom A and model \mathcal{A} of F .

(a) If “unsatisfiable” then unsatisfiable.

Assume F has model \mathcal{A} but algorithm answers “unsatisfiable”.

Let $(A_{i_1} \wedge \dots \wedge A_{i_k} \rightarrow \perp)$ be the subformula causing “unsatisfiable”.

Since $\mathcal{M}(A_{i_1}) = \dots = \mathcal{M}(A_{i_k}) = 1$, $\mathcal{A}(A_{i_1}) = \dots = \mathcal{A}(A_{i_k}) = 1$.

Then $\mathcal{A}(A_{i_1} \wedge \dots \wedge A_{i_k} \rightarrow \perp) = 0$ and so $\mathcal{A}(F) = 0$, contradiction.

(b) If “ \mathcal{M} ” then $\mathcal{M} \models F$.

After termination with “ \mathcal{M} ”, every conjunct $K \rightarrow B$ of F satisfies $\mathcal{M}(K) = 0$ or $\mathcal{M}(B) = 1$.

Therefore $\mathcal{M}(K \rightarrow B) = 1$ and thus $\mathcal{M} \models F$.

Correctness of the model building algorithm

Corollary

A satisfiable Horn formula has a unique model with a smallest number of true atoms.

Propositional Logic
DPLL: Davis-Putnam-
Logemann-Loveland

Davis–Putnam–Logemann–Loveland

DPLL algorithm:

- ▶ combines search and deduction to decide satisfiability
- ▶ underlies most modern SAT solvers
- ▶ is over 50 years old



Davis–Putnam–Logemann–Loveland

DPLL algorithm:

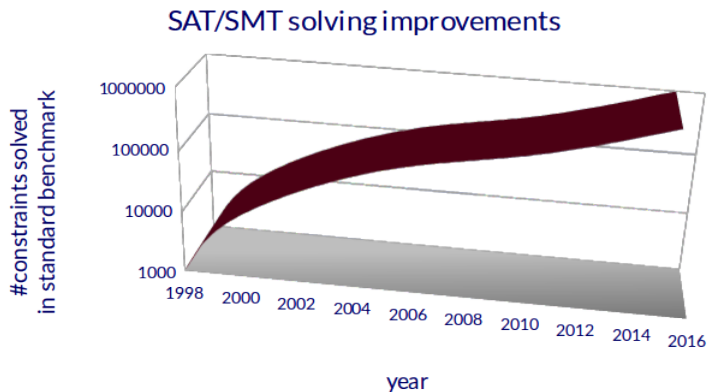
- ▶ combines search and deduction to decide satisfiability
- ▶ underlies most modern SAT solvers
- ▶ is over 50 years old



DPLL-based SAT solvers \geq 1990:

- ▶ clause learning
- ▶ non-chronological backtracking
- ▶ branching heuristics
- ▶ lazy evaluation

Performance increase of SAT solvers



Clause representation of CNF formulas

Clause representation of CNF formulas

$$\text{CNF: } (L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$$

Clause representation of CNF formulas

CNF: $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$

Representation as set of sets of literals:

$$\underbrace{\{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{k,1}, \dots, L_{1,n_k}\}\}}_{\text{clause}}$$

Clause representation of CNF formulas

CNF: $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$

Representation as set of sets of literals:

$$\underbrace{\{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{k,1}, \dots, L_{1,n_k}\}\}}_{\text{clause}}$$

Clause = set of literals (disjunction).

Clause representation of CNF formulas

CNF: $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$

Representation as set of sets of literals:

$$\underbrace{\{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{k,1}, \dots, L_{1,n_k}\}\}}_{\text{clause}}$$

Clause = set of literals (disjunction).

Formula in CNF = set of clauses

Clause representation of CNF formulas

CNF: $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$

Representation as set of sets of literals:

$$\underbrace{\{L_{1,1}, \dots, L_{1,n_1}\}}_{\text{clause}}, \dots, \{L_{k,1}, \dots, L_{1,n_k}\}$$

Clause = set of literals (disjunction).

Formula in CNF = set of clauses

Degenerate cases:

The empty clause stands for \perp .

Clause representation of CNF formulas

CNF: $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{k,1} \vee \dots \vee L_{1,n_k})$

Representation as set of sets of literals:

$$\underbrace{\{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{k,1}, \dots, L_{1,n_k}\}\}}_{\text{clause}}$$

Clause = set of literals (disjunction).

Formula in CNF = set of clauses

Degenerate cases:

The empty clause stands for \perp .

The empty set of clauses stands for \top .

The joy of sets

We get “for free”:

The joy of sets

We get “for free”:

- ▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

The joy of sets

We get “for free”:

▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

▶ **Associativity:**

$(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$

The joy of sets

We get “for free”:

▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

▶ **Associativity:**

$(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$

▶ **Idempotence:**

$(A \vee A) \equiv A$, both represented by $\{A\}$

The joy of sets

We get “for free”:

▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

▶ **Associativity:**

$(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$

▶ **Idempotence:**

$(A \vee A) \equiv A$, both represented by $\{A\}$

Sets are a convenient representation of conjunctions and disjunctions with built in associativity, commutativity and idempotence

CNF-SAT: Input: Set of clauses F

The joy of sets

We get “for free”:

▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

▶ **Associativity:**

$(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$

▶ **Idempotence:**

$(A \vee A) \equiv A$, both represented by $\{A\}$

Sets are a convenient representation of conjunctions and disjunctions with built in associativity, commutativity and idempotence

CNF-SAT: Input: Set of clauses F

Question: Is F unsatisfiable?

The joy of sets

We get “for free”:

▶ **Commutativity:**

$A \vee B \equiv B \vee A$, both represented by $\{A, B\}$

▶ **Associativity:**

$(A \vee B) \vee C \equiv A \vee (B \vee C)$, both represented by $\{A, B, C\}$

▶ **Idempotence:**

$(A \vee A) \equiv A$, both represented by $\{A\}$

Sets are a convenient representation of conjunctions and disjunctions with built in associativity, commutativity and idempotence

CNF-SAT: Input: Set of clauses F

Question: Is F unsatisfiable?

DPLL — The simplest algorithm for CNF-SAT

Simplest algorithm: Construct the truth table.

Best-case runtime is $\Theta(m \cdot 2^n)$ for a formula of length m over n variables.

DPLL — A first improvement: Partial Evaluation

Improvement: **partial evaluation** using Boole-Shannon expansion

Lemma (Boole-Shannon Expansion)

For every formula F and atom A :

$$F \equiv (A \wedge F[\top/A]) \vee (\neg A \wedge F[\perp/A]).$$

Proof By structural induction on F (exercise).

Corollary

F is satisfiable iff $F[\perp/A]$ or $F[\top/A]$ are satisfiable.

DPLL — First step: partial evaluation

$F[\perp/A]$ and $F[\top/A]$ easy to compute in clause normal form:

$F[\top/A] \equiv$ take F , remove all clauses with A , remove all $\neg A$.

$F[\perp/A] \equiv$ take F , remove all clauses with $\neg A$, remove all A .

DPLL — First step: partial evaluation

$F[\perp/A]$ and $F[\top/A]$ easy to compute in clause normal form:

$F[\top/A] \equiv$ take F , remove all clauses with A , remove all $\neg A$.

$F[\perp/A] \equiv$ take F , remove all clauses with $\neg A$, remove all A .

Partial evaluation algorithm:

Given formula F , total order on the variables \prec :

If $\{\} \in F$ return **unsatisfiable**.

If $F = \emptyset$ return **satisfiable**.

Otherwise:

Fix the first variable A in F according to \prec .

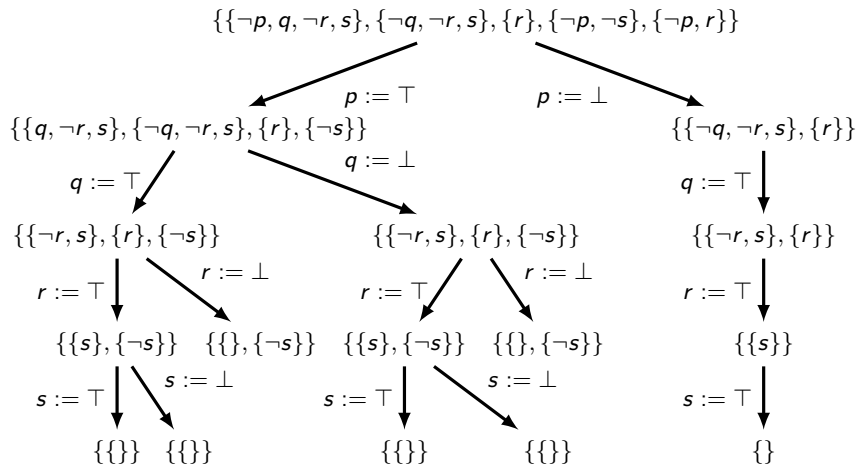
Recursively check if $F[\perp/A]$ is satisfiable;

if yes, return **satisfiable**.

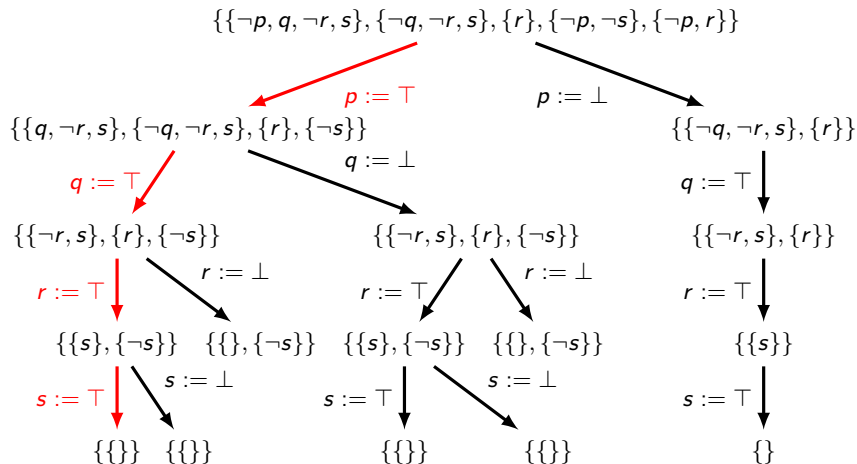
Recursively check if $F[\top/A]$ is satisfiable;

if yes, return **satisfiable**, otherwise **unsatisfiable**.

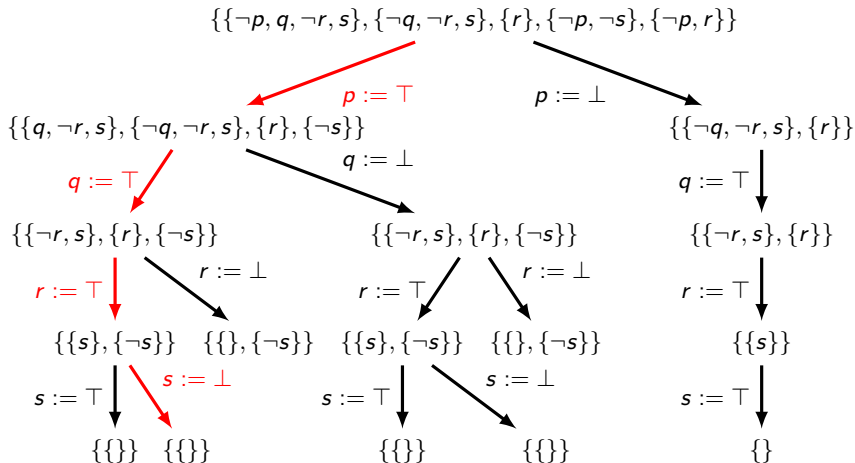
DPLL: Davis-Putnam-Logemann-Loveland



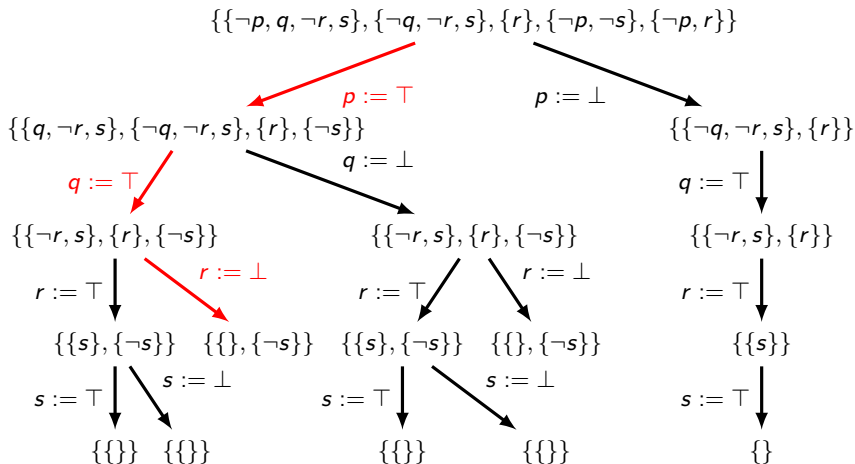
DPLL: Davis-Putnam-Logemann-Loveland



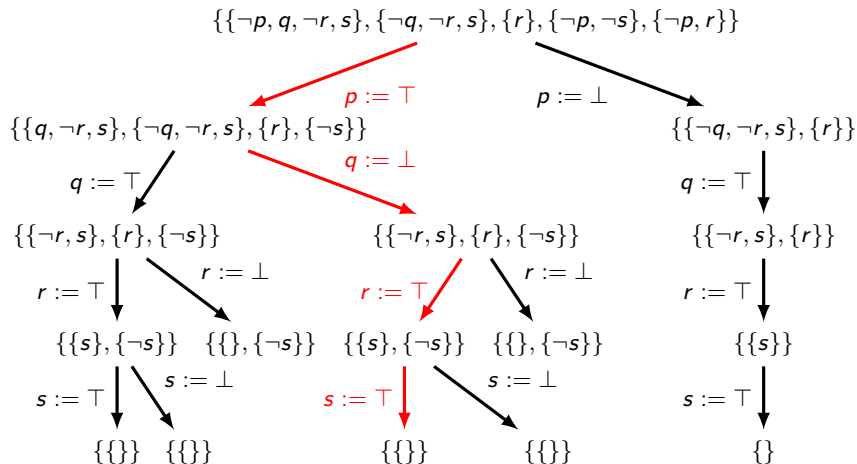
DPLL: Davis-Putnam-Logemann-Loveland



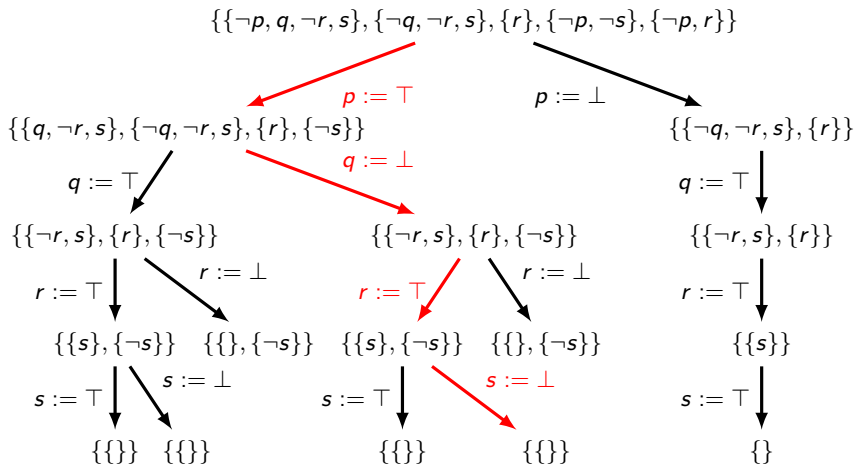
DPLL: Davis-Putnam-Logemann-Loveland



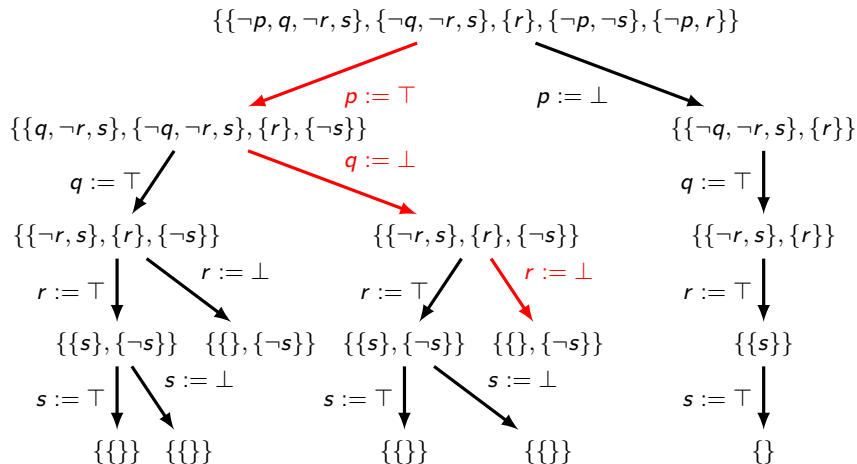
DPLL: Davis-Putnam-Logemann-Loveland



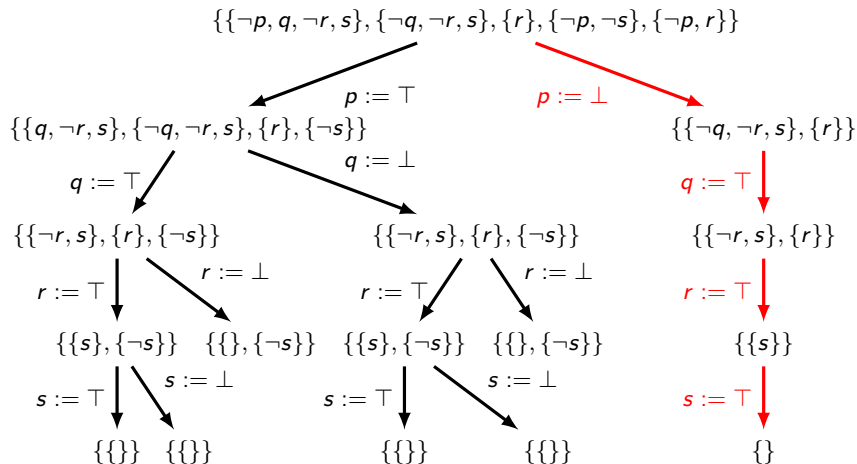
DPLL: Davis-Putnam-Logemann-Loveland



DPLL: Davis-Putnam-Logemann-Loveland



DPLL: Davis-Putnam-Logemann-Loveland



DPLL: Davis-Putnam-Logemann-Loveland

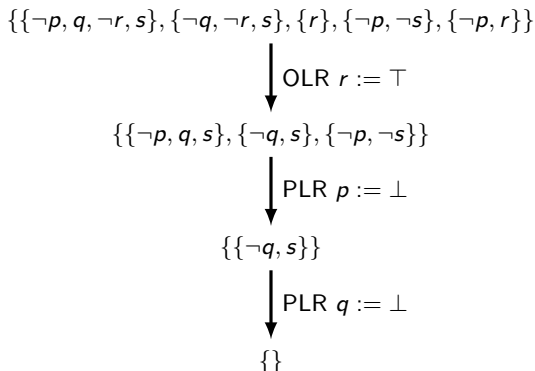
Instead of fixing an order on variables, choose the next variable **dynamically**.

- ▶ **OLR**: one-literal rule If $\{L\} \in F$ ($\{L\}$ is called **unit clause**), then every satisfying assignment sets L to true. So it suffices to check satisfiability of $F[\top/L]$.
- ▶ **PLR**: pure-literal rule
If L appears in F and \bar{L} does not, then it also suffices to check satisfiability of $F[\top/L]$ (**Why?**).

DPLL algorithm: Partial evaluation that gives priority to a variable satisfying **OLR**, then to a variable satisfying **PLR**, and otherwise picks the first unpicked variable of \prec .

Applying **OLR** can generate further unit clauses (**unit propagation**). Same for **PLR**, but DPLL often implemented with only **OLR** for efficiency.

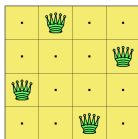
DPLL: Davis-Putnam-Logemann-Loveland



In this example PLR and OLR allow us to avoid all case splits.

Example: 4 queens

Problem: place 4 non-attacking queens on a 4x4 chess board



Variable p_{ij} models: there is a queen in square (i, j)

▶ ≥ 1 in each row: $\bigwedge_{i=1}^4 \bigvee_{j=1}^4 p_{ij}$

▶ ≤ 1 in each row: $\bigwedge_{i=1}^4 \bigwedge_{j \neq j'=1}^4 \neg p_{ij} \vee \neg p_{ij'}$

▶ ≤ 1 in each column: $\bigwedge_{j=1}^4 \bigwedge_{i \neq i'=1}^4 \neg p_{ij} \vee \neg p_{i'j}$

▶ ≤ 1 on each diagonal: $\bigwedge_{i,j=1}^4 \bigvee_k \neg p_{i-k,i+k} \vee \neg p_{i+k,j+k}$

Total number of clauses: $4 + 24 + 24 + 28 = 80$

DPLL: 4 queens

Running the DPLL algorithm:

- ▶ Start with $p_{11} \mapsto 1$
delete $\{p_{11}, p_{12}, p_{13}, p_{14}\}$, delete $\neg p_{11}$: 9 new unit clauses
unit propagation: deletes 65 clauses!

DPLL: 4 queens

Running the DPLL algorithm:

- ▶ Start with $p_{11} \mapsto 1$
delete $\{p_{11}, p_{12}, p_{13}, p_{14}\}$, delete $\neg p_{11}$: 9 new unit clauses
unit propagation: deletes 65 clauses!
- ▶ Set $p_{23} \mapsto 1$
4 new unit clauses: $\{\neg p_{24}\}, \{\neg p_{43}\}, \{\neg p_{32}\}, \{\neg p_{34}\}$
unit propagation of $\{\neg p_{34}\}$: UNSAT

DPLL: 4 queens

Running the DPLL algorithm:

- ▶ Start with $p_{11} \mapsto 1$
delete $\{p_{11}, p_{12}, p_{13}, p_{14}\}$, delete $\neg p_{11}$: 9 new unit clauses
unit propagation: deletes 65 clauses!
- ▶ Set $p_{23} \mapsto 1$
4 new unit clauses: $\{\neg p_{24}\}, \{\neg p_{43}\}, \{\neg p_{32}\}, \{\neg p_{34}\}$
unit propagation of $\{\neg p_{34}\}$: UNSAT
fixing only two literals collapsed from 80 clauses to 1
ruled out 2^{14} of 2^{16} possible assignments!
- ▶ Backtrack: $p_{11} \mapsto 0, p_{12} \mapsto 1$
delete $\{\neg p_{12}\}$: 9 new unit clauses
unit propagation: leaves only 1 clause $\{p_{43}\}$!

DPLL: 4 queens

Running the DPLL algorithm:

- ▶ Start with $p_{11} \mapsto 1$
delete $\{p_{11}, p_{12}, p_{13}, p_{14}\}$, delete $\neg p_{11}$: 9 new unit clauses
unit propagation: deletes 65 clauses!
- ▶ Set $p_{23} \mapsto 1$
4 new unit clauses: $\{\neg p_{24}\}, \{\neg p_{43}\}, \{\neg p_{32}\}, \{\neg p_{34}\}$
unit propagation of $\{\neg p_{34}\}$: UNSAT
fixing only two literals collapsed from 80 clauses to 1
ruled out 2^{14} of 2^{16} possible assignments!
- ▶ Backtrack: $p_{11} \mapsto 0$, $p_{12} \mapsto 1$
delete $\{\neg p_{12}\}$: 9 new unit clauses
unit propagation: leaves only 1 clause $\{p_{43}\}$!
- ▶ Answer: $p_{12}, p_{24}, p_{31}, p_{43} \mapsto 1$

DPLL: Evaluation

Oriented towards satisfiability:

- ▶ $2^{O(n)}$ time for satisfiable formulas, but $2^{\Theta(n)}$ for unsatisfiable ones.
- ▶ DPLL computes a satisfying assignment, if there is one.
- ▶ The satisfying assignment is a **certificate** of satisfiability.
- ▶ Satisfiable formulas have short certificates: satisfying assignment never larger than the formula.

Coming next: **resolution**, a procedure oriented towards unsatisfiability.

- ▶ $2^{O(n)}$ time for unsatisfiable formulas, but $2^{\Theta(n)}$ for satisfiable ones.
- ▶ Resolution computes a certificate of unsatisfiability.
- ▶ However, the certificate is **exponentially longer** than the formula in the worst case.
- ▶ Polynomial certificates for satisfiability implies $NP = coNP$.

Propositional Logic

Compactness

Compactness Theorem

Theorem

*A set S of formulas is satisfiable
iff every finite subset of S is satisfiable.*

Equivalent formulation:

*A set S of formulas is unsatisfiable
iff some finite subset of S is unsatisfiable.*

An application: Graph Coloring

Definition

A **4-coloring** of a graph (V, E) is a map $c : V \rightarrow \{1, 2, 3, 4\}$ such that $(x, y) \in E$ implies $c(x) \neq c(y)$.

Theorem (4CT)

A finite planar graph has a 4-coloring.

Theorem

A planar graph $G = (V, E)$ with countably many vertices $V = \{v_1, v_2, \dots\}$ has a 4-coloring.

Proof $G \rightsquigarrow$ set of formulas S s.t. S is sat. iff G is 4-col.

G is planar

\Rightarrow every finite subgraph of G is planar and 4-col. (by 4CT)

\Rightarrow every finite subset of S is sat.

$\Rightarrow S$ is sat. (by Compactness)

$\Rightarrow G$ is 4-col.

Proof details

$G \rightsquigarrow S$:

For simplicity:

atoms are of the form A_i^c where $c \in \{1, \dots, 4\}$ and $i \in \mathbb{N}$

$$S := \{A_i^1 \vee A_i^2 \vee A_i^3 \vee A_i^4 \mid i \in \mathbb{N}\} \cup \\ \{A_i^c \rightarrow \neg A_i^d \mid i \in \mathbb{N}, c, d \in \{1, \dots, 4\}, c \neq d\} \cup \\ \{\neg(A_i^c \wedge A_j^c) \mid (v_i, v_j) \in E, c \in \{1, \dots, 4\}\}$$

Subgraph corresponding to some $T \subseteq S$:

$$V_T := \{v_i \mid A_i^c \text{ occurs in } T \text{ (for some } c)\}$$

$$E_T := \{(v_i, v_j) \mid \neg(A_i^c \wedge A_j^c) \in T \text{ (for some } c)\}$$

Proof of Compactness

Theorem

*A set S of formulas is satisfiable
iff every finite subset of S is satisfiable.*

Proof

\Rightarrow : If S is satisfiable then every finite subset of S is satisfiable.

Trivial.

\Leftarrow : If every finite subset of S is satisfiable then S is satisfiable.

We prove that S has a model.

Proof of Compactness

Definition

Let $b_1 \cdots b_n \in \{0, 1\}^*$ with $n \geq 0$ and let T be a set of formulas. An assignment \mathcal{A} is a $b_1 \cdots b_n$ -model of T if $\mathcal{A}(A_i) = b_i$ for every $i = 1, \dots, n$ and $\mathcal{A} \models T$.

In particular: every model is a ε -model.

Assume every finite $T \subseteq S$ is satisfiable. We prove:

1. There is an infinite sequence $b_1 b_2 \cdots \in \{0, 1\}^\omega$ such that for every $n \geq 1$ all finite $T \subseteq S$ have a $b_1 \cdots b_n$ -model.
2. The assignment \mathcal{B} given by $\mathcal{B}(A_i) := b_i$ for all $i \geq 1$ is a model of S .

Proof of Compactness: Part (1)

To prove: There is an infinite sequence $b_1 b_2 \cdots \in \{0, 1\}^\omega$ such that for every $n \geq 1$ all finite $T \subseteq S$ have a $b_1 \cdots b_n$ -model.

It suffices to show:

- (a) Every finite $T \subseteq S$ has an ε -model.
- (b) For every sequence $\sigma \in \{0, 1\}^*$: if every finite $T \subseteq S$ has a σ -model then there exists $b \in \{0, 1\}$ such that every finite $T \subseteq S$ has a σb -model.

Proof of (a): By assumption every $T \subseteq S$ has an ε -model.

Proof of (b): Next slide.

Proof of Compactness: Part (1)

Proof of (b): By contradiction.

Assume that for some $\sigma \in \{0, 1\}$: every finite $T \subseteq S$ has a σ -model, but

- (0) some finite $T_0 \subseteq S$ has no $\sigma 0$ -model; and
- (1) some finite $T_1 \subseteq S$ has no $\sigma 1$ -model.

Consider the finite set $T_0 \cup T_1$.

By assumption, $T_0 \cup T_1$ has some σ -model \mathcal{A} . Let $n := |\sigma|$.

Two possible cases:

- ▶ $\mathcal{A}(A_{n+1}) = 0$. Then \mathcal{A} is a $\sigma 0$ -model of T_0 , contradicting (0).
- ▶ $\mathcal{A}(A_{n+1}) = 1$. Then \mathcal{A} is a $\sigma 1$ -model of T_1 , contradicting (1).

Proof of Compactness: Part (2)

To prove: The assignment \mathcal{B} given by $\mathcal{B}(A_i) := b_i$ for all $i \geq 1$, where $b_1 b_2 \dots$ is the infinite sequence of (1), is a model of S .

We show $\mathcal{B} \models F$ for all $F \in S$.

Let m be the maximal index of all atoms in F .

By (1), $\{F\}$ has a $b_1 \dots b_m$ -model \mathcal{A} .

Hence $\mathcal{B} \models F$, because \mathcal{A} and \mathcal{B} agree on all atoms in F .

Corollary

Corollary

If $S \models F$ then there is a finite subset $M \subseteq S$ such that $M \models F$.

Propositional Logic Resolution

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Algorithm:

Keep on “resolving” two clauses from F and adding the result to F

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Algorithm:

Keep on “resolving” two clauses from F and adding the result to F until the empty clause is found

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Algorithm:

Keep on “resolving” two clauses from F and adding the result to F until the empty clause is found

Correctness:

If the empty clause is found, the initial F is unsatisfiable

Completeness:

If the initial F is unsatisfiable, the empty clause can be found.

Resolution — The idea

Input: Set of clauses F

Question: Is F unsatisfiable?

Algorithm:

Keep on “resolving” two clauses from F and adding the result to F until the empty clause is found

Correctness:

If the empty clause is found, the initial F is unsatisfiable

Completeness:

If the initial F is unsatisfiable, the empty clause can be found.

Correctness/Completeness of **syntactic** procedure (**resolution**)
w.r.t. **semantic** property (**unsatisfiability**)

Resolvent

Definition

Let L be a literal. Then \bar{L} is defined as follows:

$$\bar{L} = \begin{cases} \neg A_i & \text{if } L = A_i \\ A_i & \text{if } L = \neg A_i \end{cases}$$

Resolvent

Definition

Let L be a literal. Then \bar{L} is defined as follows:

$$\bar{L} = \begin{cases} \neg A_i & \text{if } L = A_i \\ A_i & \text{if } L = \neg A_i \end{cases}$$

Definition

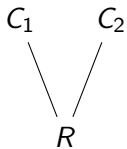
Let C_1, C_2 be clauses and let L be a literal such that $L \in C_1$ and $\bar{L} \in C_2$. Then the clause

$$(C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$$

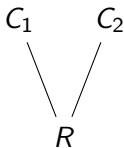
is a **resolvent** of C_1 and C_2 .

The process of deriving the resolvent is called a **resolution step**.

Graphical representation of resolvent:



Graphical representation of resolvent:



If $C_1 = \{L\}$ and $C_2 = \{\bar{L}\}$ then the empty clause is a resolvent of C_1 and C_2 . The special symbol \square denotes the empty clause.

Recall: \square represents \perp .

Resolution proof

Definition

A **resolution proof** of a clause C from a set of clauses F is a sequence of clauses C_0, \dots, C_n such that

- ▶ $C_i \in F$ or C_i is a resolvent of two clauses C_a and C_b , $a, b < i$,
- ▶ $C_n = C$

Then we can write $F \vdash_{Res} C$.

Note: F can be finite or infinite!

Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

Example

$\{P, Q\}$

$\{P, \neg Q\}$

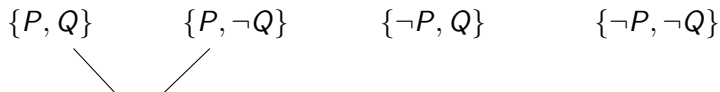
$\{\neg P, Q\}$

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

Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

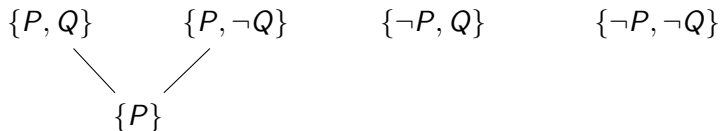
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

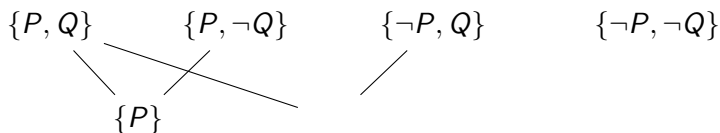
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

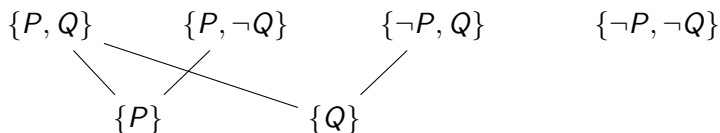
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

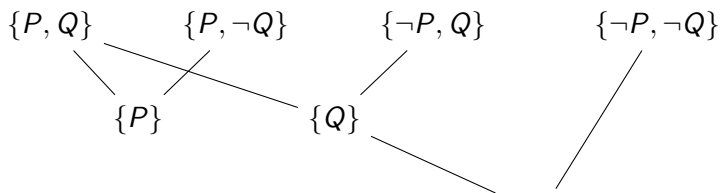
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

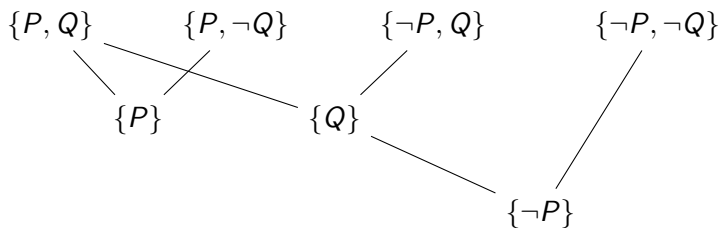
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

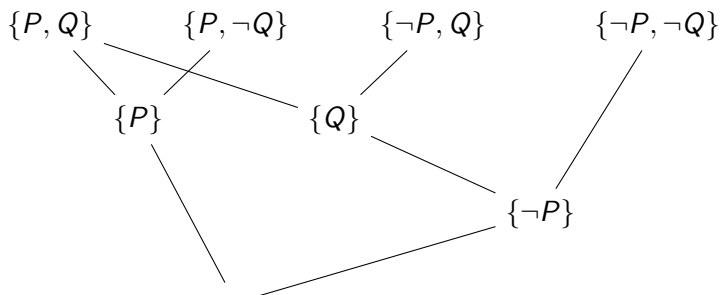
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

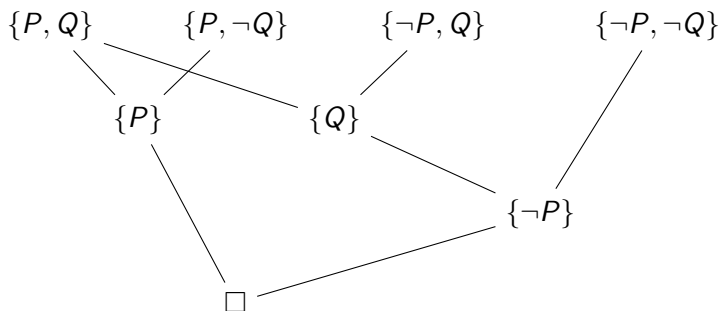
Example



Resolution proof as DAG

A resolution proof can be shown as a DAG with the clauses in F as the leaves and C as the root:

Example



A linear resolution proof

0: $\{P, Q\}$

1: $\{P, \neg Q\}$

2: $\{\neg P, Q\}$

3: $\{\neg P, \neg Q\}$

4: $\{P\}$ (0, 1)

5: $\{Q\}$ (0, 2)

6: $\{\neg P\}$ (3, 5)

7: \square (4, 6)

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 .

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 . Then $C_1, C_2 \models R$.

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 . Then $C_1, C_2 \models R$.

Proof By definition $R = (C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$ (for some L).

Assume $\mathcal{A} \models C_1$ and $\mathcal{A} \models C_2$. We show $\mathcal{A} \models R$.

There are two cases:

- ▶ $\mathcal{A} \models L$. Then $\mathcal{A} \models C_2 - \{\bar{L}\}$ (because $\mathcal{A} \models C_2$),

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 . Then $C_1, C_2 \models R$.

Proof By definition $R = (C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$ (for some L).

Assume $\mathcal{A} \models C_1$ and $\mathcal{A} \models C_2$. We show $\mathcal{A} \models R$.

There are two cases:

- ▶ $\mathcal{A} \models L$. Then $\mathcal{A} \models C_2 - \{\bar{L}\}$ (because $\mathcal{A} \models C_2$), thus $\mathcal{A} \models R$.
- ▶ $\mathcal{A} \not\models L$. Then $\mathcal{A} \models C_1 - \{L\}$ (because $\mathcal{A} \models C_1$),

Correctness of resolution

Lemma (Resolution Lemma)

Let R be a resolvent of two clauses C_1 and C_2 . Then $C_1, C_2 \models R$.

Proof By definition $R = (C_1 - \{L\}) \cup (C_2 - \{\bar{L}\})$ (for some L).

Assume $\mathcal{A} \models C_1$ and $\mathcal{A} \models C_2$. We show $\mathcal{A} \models R$.

There are two cases:

- ▶ $\mathcal{A} \models L$. Then $\mathcal{A} \models C_2 - \{\bar{L}\}$ (because $\mathcal{A} \models C_2$), thus $\mathcal{A} \models R$.
- ▶ $\mathcal{A} \not\models L$. Then $\mathcal{A} \models C_1 - \{L\}$ (because $\mathcal{A} \models C_1$), thus $\mathcal{A} \models R$.

Correctness of resolution

Theorem (Correctness of resolution)

Let F be a set of clauses. If $F \vdash_{Res} C$ then $F \models C$.

Correctness of resolution

Theorem (Correctness of resolution)

Let F be a set of clauses. *If $F \vdash_{Res} C$ then $F \models C$.*

Proof Assume there is a resolution proof $C_0, \dots, C_n = C$.
We show $F \models C_i$ by induction on i . IH: $F \models C_j$ for all $j < i$.

Correctness of resolution

Theorem (Correctness of resolution)

Let F be a set of clauses. If $F \vdash_{Res} C$ then $F \models C$.

Proof Assume there is a resolution proof $C_0, \dots, C_n = C$.

We show $F \models C_i$ by induction on i . IH: $F \models C_j$ for all $j < i$.

There are two cases:

- ▶ $C_i \in F$.

Then $F \models C_i$ by definition.

- ▶ C_i is a resolvent of C_a and C_b for $a, b < i$.

Then $F \models C_a$ and $F \models C_b$ by IH, and $C_a, C_b \models C_i$ by the resolution lemma. Thus $F \models C_i$.

Correctness of resolution

Theorem (Correctness of resolution)

Let F be a set of clauses. *If $F \vdash_{Res} C$ then $F \models C$.*

Proof Assume there is a resolution proof $C_0, \dots, C_n = C$. We show $F \models C_i$ by induction on i . IH: $F \models C_j$ for all $j < i$. There are two cases:

- ▶ $C_i \in F$.
Then $F \models C_i$ by definition.
- ▶ C_i is a resolvent of C_a and C_b for $a, b < i$.
Then $F \models C_a$ and $F \models C_b$ by IH, and $C_a, C_b \models C_i$ by the resolution lemma. Thus $F \models C_i$.

Corollary

Let F be a set of clauses. *If $F \vdash_{Res} \square$ then F is unsatisfiable.*

Completeness of resolution

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Theorem (Completeness of resolution)

Let F be a set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Completeness of resolution

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Theorem (Completeness of resolution)

Let F be a set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Proof If F is infinite, there must be a finite unsatisfiable subset of F (by the Compactness Theorem);

Completeness of resolution

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Theorem (Completeness of resolution)

Let F be a set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Proof If F is infinite, there must be a finite unsatisfiable subset of F (by the Compactness Theorem); in that case let F be that finite subset and apply the previous theorem.

Completeness of resolution

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Theorem (Completeness of resolution)

Let F be a set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Proof If F is infinite, there must be a finite unsatisfiable subset of F (by the Compactness Theorem); in that case let F be that finite subset and apply the previous theorem.

Corollary

A set of clauses F is unsatisfiable iff $F \vdash_{Res} \square$.

Completeness proof

Corollary

(of the Boole-Shannon expansion) F is unsatisfiable iff $F[\perp/A]$ and $F[\top/A]$ are unsatisfiable.

Idea for completeness proof:

If A is an atom of F , then both $F[\perp/A]$ and $F[\top/A]$ have fewer atoms than F .

Use Boole-Shannon to prove completeness by induction on the number of atoms of the unsatisfiable formula F :

- ▶ construct inductively resolution proofs for $F[\perp/A]$ and $F[\top/A]$, and
- ▶ “combine” them into a resolution proof for F .

Inductive construction of resolution proofs

$$F = \{ \{ \neg q, s \}, \{ \neg p, q, s \}, \{ p \}, \{ r, \neg s \}, \{ \neg p, \neg r, \neg s \} \}$$

- ▶ Compute inductively proofs for $F[\top/s]$ and $F[\perp/s]$.

$$F[\top/s] \equiv \{ \{ p \}, \{ r \}, \{ \neg p, \neg r \} \}$$

$$F[\perp/s] \equiv \{ \{ \neg q \}, \{ \neg p, q \}, \{ p \} \}$$

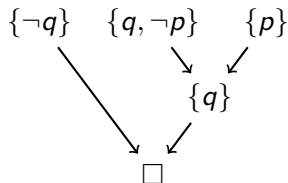
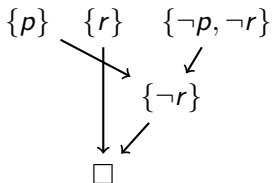
Inductive construction of resolution proofs

$$F = \{ \{ \neg q, s \}, \{ \neg p, q, s \}, \{ p \}, \{ r, \neg s \}, \{ \neg p, \neg r, \neg s \} \}$$

- Compute inductively proofs for $F[\top/s]$ and $F[\perp/s]$.

$$F[\top/s] \equiv \{ \{ p \}, \{ r \}, \{ \neg p, \neg r \} \}$$

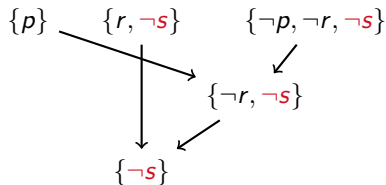
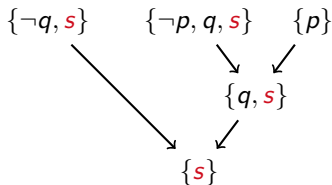
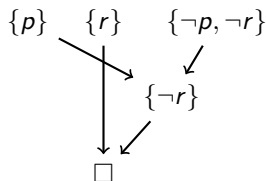
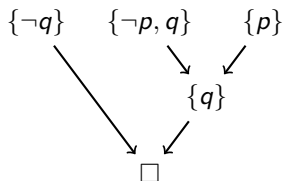
$$F[\perp/s] \equiv \{ \{ \neg q \}, \{ \neg p, q \}, \{ p \} \}$$



Inductive construction of resolution proofs

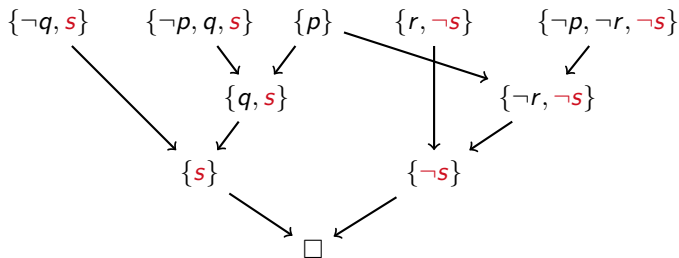
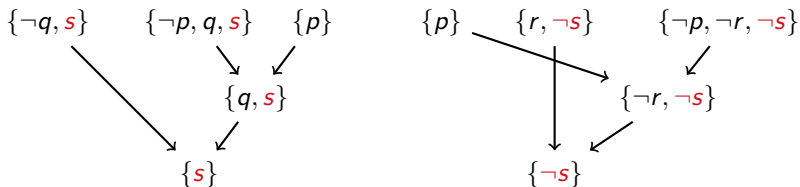
- ▶ Reintroduce s and $\neg s$.

$$F = \{ \{ \neg q, s \}, \{ \neg p, q, s \}, \{ p \}, \{ r, \neg s \}, \{ \neg p, \neg r, \neg s \} \}$$



Inductive construction of resolution proofs

- Combine the graphs for $\{s\}$ and $\{\neg s\}$.



Completeness proof

Theorem

Let F be a finite set of clauses. If F is unsatisfiable then $F \vdash_{Res} \square$.

Completeness proof

Theorem

Let F be a finite set of clauses. *If F is unsatisfiable then $F \vdash_{Res} \square$.*

Proof By induction on the number n of distinct atoms in F .

Basis: If $n = 0$ then $F = \{\}$ (but F is unsat.)

Completeness proof

Theorem

Let F be a finite set of clauses. *If F is unsatisfiable then $F \vdash_{Res} \square$.*

Proof By induction on the number n of distinct atoms in F .

Basis: If $n = 0$ then $F = \{\}$ (but F is unsat.) or $F = \{\square\}$.

Step:

IH: For every unsat. set of clauses F with n dist. atoms, $F \vdash_{Res} \square$.

Let F contain $n + 1$ distinct atoms. Pick some atom A in F .

$F[\top/A] \equiv$ take F , remove all clauses with A , remove all $\neg A$.

$F[\perp/A] \equiv$ take F , remove all clauses with $\neg A$, remove all A .

Completeness proof

By IH: there are res. proofs $C_0, \dots, C_m = \square$ from $F[\perp/A]$ and $D_0, \dots, D_n = \square$ from $F[\top/A]$.

Now transform C_0, \dots, C_m into a proof C'_0, \dots, C'_m from F by adding A back into the clauses it was removed from. Then:

- ▶ either $C'_m = \{A\}$
- ▶ or $C'_m = \square$ (and we are done).

Similarly we transform D_0, \dots, D_n into a proof D'_0, \dots, D'_n from F by adding $\neg A$ back in. Then:

- ▶ either $D'_n = \{\neg A\}$
- ▶ or $D'_n = \square$ (and we are done).

If $C'_m = \{A\}$ and $D'_n = \{\neg A\}$ then $F \vdash_{Res} A$ and $F \vdash_{Res} \neg A$ and thus $F \vdash_{Res} \square$.

Resolution is only refutation complete

Not everything that is a consequence of a set of clauses
can be derived by resolution.

Resolution is only refutation complete

Not everything that is a consequence of a set of clauses
can be derived by resolution.

Exercise

Find F and C such that $F \models C$ but not $F \vdash_{Res} C$.

Resolution is only refutation complete

Not everything that is a consequence of a set of clauses
can be derived by resolution.

Exercise

Find F and C such that $F \models C$ but not $F \vdash_{Res} C$.

How to prove $F \models C$ by resolution?

Resolution is only refutation complete

Not everything that is a consequence of a set of clauses
can be derived by resolution.

Exercise

Find F and C such that $F \models C$ but not $F \vdash_{Res} C$.

How to prove $F \models C$ by resolution?

Prove $F \cup \{\neg C\} \vdash_{Res} \square$

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

while there are clauses $C_a, C_b \in F$ and resolvent R of C_a and C_b
such that $R \notin F$
do $F := F \cup \{R\}$

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

while there are clauses $C_a, C_b \in F$ and resolvent R of C_a and C_b
such that $R \notin F$
do $F := F \cup \{R\}$

Lemma

The algorithm terminates.

Proof There are only finitely many clauses over a finite set of atoms.

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

while there are clauses $C_a, C_b \in F$ and resolvent R of C_a and C_b
such that $R \notin F$
do $F := F \cup \{R\}$

Lemma

The algorithm terminates.

Proof There are only finitely many clauses over a finite set of atoms.

Theorem

The initial F is unsatisfiable iff \square is in the final F

Proof F_{init} is unsat. iff $F_{init} \vdash_{Res} \square$ iff $\square \in F_{final}$ because the algorithm enumerates all R such that $F_{init} \vdash_{Res} R$.

A resolution algorithm

Input: A CNF formula F , i.e. a finite set of clauses

while there are clauses $C_a, C_b \in F$ and resolvent R of C_a and C_b
such that $R \notin F$
do $F := F \cup \{R\}$

Lemma

The algorithm terminates.

Proof There are only finitely many clauses over a finite set of atoms.

Theorem

The initial F is unsatisfiable iff \square is in the final F

Proof F_{init} is unsat. iff $F_{init} \vdash_{Res} \square$ iff $\square \in F_{final}$ because the algorithm enumerates all R such that $F_{init} \vdash_{Res} R$.

The algorithm is a decision procedure for unsat. of CNF formulas.

Propositional Logic
CDCL: Conflict Driven Clause
Learning

CDCL: goal and idea

Goal: Combine DPLL and resolution into an algorithm oriented towards both satisfiability and unsatisfiability.

Idea: At every unsuccessful leaf of DPLL (called **conflict**), compute a **conflict clause**, and add it to the formula we are deciding about.

Conflict clauses “cache” previous search results, so we “learn from previous mistakes”.

Conflict clauses also determine backtracking.

We present a particular way of computing a conflict clause using resolution. There are other ways.

DPLL + CDCL algorithm

Given formula F and **partial assignment** \mathcal{A} :

$F|_{\mathcal{A}}$ denotes the result of deleting any clause containing a true literal, and deleting all false literals from each remaining clause.

Input: CNF formula F .

1. Initialise \mathcal{A} to the empty assignment
2. While there is unit clause $\{L\}$ or pure literal L in $F|_{\mathcal{A}}$, update $\mathcal{A} \mapsto \mathcal{A}[\top/L]$
3. If $F|_{\mathcal{A}} = \emptyset$, stop and output \mathcal{A} .
4. If $F|_{\mathcal{A}} \ni \square$, add new clause C to F by **learning procedure**.
If $C = \square$, stop and output UNSAT; otherwise backtrack to highest level where C is unit clause.
Go to line 2.
5. Apply **decision strategy** to update \mathcal{A} .
Go to line 2.

Terminology

- ▶ **State** of algorithm is pair (F, \mathcal{A}) , where F is CNF formula and \mathcal{A} is partial assignment.
Successful state when $\mathcal{A} \models F$. **Conflict state** when $\mathcal{A} \not\models F$.
(Note: conflict state if $F|_{\mathcal{A}} \ni \square$, successful state if $F|_{\mathcal{A}} = \emptyset$)
- ▶ Each assignment $A_i \mapsto b_i$ classifies as **decision assignment** or **implied assignment**.
- ▶ $A_i \mapsto b_i$ denotes decision assignment with **decision variable** A_i .
- ▶ $A_i \xrightarrow{C} b_i$ denotes an implied assignment arising through **unit propagation** on clause C .

Terminology

- ▶ **State** of algorithm is pair (F, \mathcal{A}) , where F is CNF formula and \mathcal{A} is partial assignment.
Successful state when $\mathcal{A} \models F$. **Conflict state** when $\mathcal{A} \not\models F$.
(Note: conflict state if $F|_{\mathcal{A}} \ni \square$, successful state if $F|_{\mathcal{A}} = \emptyset$)
- ▶ Each assignment $A_i \mapsto b_i$ classifies as **decision assignment** or **implied assignment**.
- ▶ $A_i \mapsto b_i$ denotes decision assignment with **decision variable** A_i .
- ▶ $A_i \xrightarrow{C} b_i$ denotes an implied assignment arising through **unit propagation** on clause C .
- ▶ **Decision level** of assignment $A_i \mapsto b_i$ in a given state (F, \mathcal{A}) is number of decision assignments in \mathcal{A} that precede $A_i \mapsto b_i$.

Example: start with set of clauses $F = \{C_1, \dots, C_5\}$, where

$$C_1 = \{\neg A_1, \neg A_4, A_5\}$$

$$C_2 = \{\neg A_1, A_6, \neg A_5\}$$

$$C_3 = \{\neg A_1, \neg A_6, A_7\}$$

$$C_4 = \{\neg A_1, \neg A_7, \neg A_5\}$$

$$C_5 = \{A_1, A_4, A_6\}$$

Say current assignment is $(A_1 \mapsto 1, A_2 \mapsto 0, A_3 \mapsto 0, A_4 \mapsto 1)$.

Notice $F|_{\mathcal{A}}$ contains unit clause $\{A_5\}$.

Unit propagation further generates $(A_5 \xrightarrow{C_1} 1, A_6 \xrightarrow{C_2} 1, A_7 \xrightarrow{C_3} 1)$.

This leads to a conflict, with C_4 being made false.

Conflict analysis

After unit propagation:

- ▶ If not in conflict nor successful, make decision (line 5)
- ▶ If in conflict, **learned clause** is added (line 4)

Conflict analysis

After unit propagation:

- ▶ If not in conflict nor successful, make decision (line 5)
- ▶ If in conflict, **learned clause** is added (line 4)

Learned clause desiderata: If unit propagation from state (F, \mathcal{A}) leads to conflict, clause C is learned such that:

Conflict analysis

After unit propagation:

- ▶ If not in conflict nor successful, make decision (line 5)
- ▶ If in conflict, **learned clause** is added (line 4)

Learned clause desiderata: If unit propagation from state (F, \mathcal{A}) leads to conflict, clause C is learned such that:

1. $F \equiv F \cup \{C\}$

Conflict analysis

After unit propagation:

- ▶ If not in conflict nor successful, make decision (line 5)
- ▶ If in conflict, **learned clause** is added (line 4)

Learned clause desiderata: If unit propagation from state (F, \mathcal{A}) leads to conflict, clause C is learned such that:

1. $F \equiv F \cup \{C\}$
2. C is **conflict clause**: each literal of C is made false by \mathcal{A}

Conflict analysis

After unit propagation:

- ▶ If not in conflict nor successful, make decision (line 5)
- ▶ If in conflict, **learned clause** is added (line 4)

Learned clause desiderata: If unit propagation from state (F, \mathcal{A}) leads to conflict, clause C is learned such that:

1. $F \equiv F \cup \{C\}$
2. C is **conflict clause**: each literal of C is made false by \mathcal{A}
3. C mentions only decision variables in \mathcal{A}

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \dots, A_k \mapsto b_k)$ leads to conflict.

Find associated clauses D_1, \dots, D_{k+1} by backward induction:

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \dots, A_k \mapsto b_k)$ leads to conflict.

Find associated clauses D_1, \dots, D_{k+1} by backward induction:

1. $D_{k+1} :=$ any conflict clause of F under \mathcal{A} .

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \dots, A_k \mapsto b_k)$ leads to conflict.

Find associated clauses D_1, \dots, D_{k+1} by backward induction:

1. $D_{k+1} :=$ any conflict clause of F under \mathcal{A} .
2. If $A_i \mapsto b_i$ is decision assignment or A_i not mentioned in D_{i+1} , set $D_i := D_{i+1}$.

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \dots, A_k \mapsto b_k)$ leads to conflict.

Find associated clauses D_1, \dots, D_{k+1} by backward induction:

1. $D_{k+1} :=$ any conflict clause of F under \mathcal{A} .
2. If $A_i \mapsto b_i$ is decision assignment or A_i not mentioned in D_{i+1} , set $D_i := D_{i+1}$.
3. If $A_i \stackrel{C_i}{\mapsto} b_i$ is implied assignment and A_i mentioned in D_{i+1} , define D_i to be resolvent of D_{i+1} and C_i with respect to A_i .

Clause learning using resolution

Suppose $\mathcal{A} = (A_1 \mapsto b_1, \dots, A_k \mapsto b_k)$ leads to conflict.

Find associated clauses D_1, \dots, D_{k+1} by backward induction:

1. $D_{k+1} :=$ any conflict clause of F under \mathcal{A} .
2. If $A_i \mapsto b_i$ is decision assignment or A_i not mentioned in D_{i+1} , set $D_i := D_{i+1}$.
3. If $A_i \stackrel{C_i}{\mapsto} b_i$ is implied assignment and A_i mentioned in D_{i+1} , define D_i to be resolvent of D_{i+1} and C_i with respect to A_i .

$C := A_1$, that is, the final clause A_1 is the **learned clause** .

Clause learning: example

Conflict of example above:

$$\begin{array}{ll} C_1 = \{\neg A_1, \neg A_4, A_5\} & D_8 := \{\neg A_1, \neg A_7, \neg A_5\} \quad (\text{clause } C_4) \\ C_2 = \{\neg A_1, A_6, \neg A_5\} & D_7 := \{\neg A_1, \neg A_5, \neg A_6\} \quad (\text{resolve } D_8, C_3) \\ C_3 = \{\neg A_1, \neg A_6, A_7\} & D_6 := \{\neg A_1, \neg A_5\} \quad (\text{resolve } D_7, C_2) \\ C_4 = \{\neg A_1, \neg A_7, \neg A_5\} & D_5 := \{\neg A_1, \neg A_4\} \quad (\text{resolve } D_6, C_1) \\ C_5 = \{A_1, A_4, A_6\} & D_4 := \{\neg A_1, \neg A_4\} \\ A_1 \mapsto 1, A_2 \mapsto 0, & D_3 := \{\neg A_1, \neg A_4\} \\ A_3 \mapsto 0, A_4 \mapsto 1, & D_2 := \{\neg A_1, \neg A_4\} \\ A_5 \stackrel{C_1}{\mapsto} 1, A_6 \stackrel{C_2}{\mapsto} 1, & D_1 := \{\neg A_1, \neg A_4\} \\ A_7 \stackrel{C_3}{\mapsto} 1 & \end{array}$$

Clause learning: example

Conflict of example above:

$$\begin{array}{ll} C_1 = \{\neg A_1, \neg A_4, A_5\} & D_8 := \{\neg A_1, \neg A_7, \neg A_5\} \quad (\text{clause } C_4) \\ C_2 = \{\neg A_1, A_6, \neg A_5\} & D_7 := \{\neg A_1, \neg A_5, \neg A_6\} \quad (\text{resolve } D_8, C_3) \\ C_3 = \{\neg A_1, \neg A_6, A_7\} & D_6 := \{\neg A_1, \neg A_5\} \quad (\text{resolve } D_7, C_2) \\ C_4 = \{\neg A_1, \neg A_7, \neg A_5\} & D_5 := \{\neg A_1, \neg A_4\} \quad (\text{resolve } D_6, C_1) \\ C_5 = \{A_1, A_4, A_6\} & D_4 := \{\neg A_1, \neg A_4\} \\ A_1 \mapsto 1, A_2 \mapsto 0, & D_3 := \{\neg A_1, \neg A_4\} \\ A_3 \mapsto 0, A_4 \mapsto 1, & D_2 := \{\neg A_1, \neg A_4\} \\ A_5 \stackrel{C_1}{\mapsto} 1, A_6 \stackrel{C_2}{\mapsto} 1, & D_1 := \{\neg A_1, \neg A_4\} \\ A_7 \stackrel{C_3}{\mapsto} 1 & \end{array}$$

Learned clause D_1 is conflict clause with only decision variables, including top-level one A_1 .

Clause learning: example

Intuitively:

- ▶ D_1 records that conflict due to decision to make A_1, A_4 true.
- ▶ Adding D_1 ensures search does not explore assignments with $A_1 \mapsto 1, A_4 \mapsto 1$.
- ▶ DPLL backtracks to highest level where D_1 is unit clause (after $A_1 \mapsto 1$), unit propagation leads to $A_4 \mapsto 0$.

Clause learning

Proposition: The clause learning procedure satisfies the three desiderata.

Proof sketch: **Observation:** If $A_i \stackrel{C_i}{\mapsto} b_i$, then the only literal of C_i true under \mathcal{A} is the literal for A_i (that is, C_i contains either A_i or $\neg A_i$, and b_i is chosen to make the literal true).

1. $F \equiv F \cup \{C\}$

Because C is obtained from clauses of F through resolution steps.

2. C is **conflict clause**: each literal is made false by \mathcal{A} .

We show by induction that $D_{k+1}, D_k, \dots, D_1 = C$ are conflict clauses.

D_{k+1} is conflict clause by definition.

If D_{i+1} is conflict clause and $D_i = D_{i+1}$, then so is D_i .

If D_{i+1} is conflict clause and $D_i \neq D_{i+1}$, then D_i is the result of resolving D_{i+1} and C_i . By the **observation**, all literals of D_i are made false by \mathcal{A} .

3. C mentions only decision variables in \mathcal{A} .

Because every other variable, say A_i , disappears after resolving with D_{i+1} w.r.t. A_i .

Indeed, since \mathcal{A} makes D_{i+1} false, by the **observation** A_i has opposite signs in D_{i+1} and C_i .

Example (without PLR)

$$\{\neg A_1\} \{A_1, A_3, A_4\} \{\neg A_2, \neg A_5\} \{A_3, \neg A_4, A_5, \neg A_6\} \{A_1, \neg A_2, \neg A_4, A_6\}$$

$$\text{OLR: } A_1 \mapsto 0 \quad \{A_3, A_4\} \quad \{\neg A_2, \neg A_5\} \quad \{A_3, \neg A_4, A_5, \neg A_6\} \quad \{\neg A_2, \neg A_4, A_6\}$$

$$\text{DE: } A_2 \mapsto 1 \quad \{A_3, A_4\} \quad \{\neg A_5\} \quad \{A_3, \neg A_4, A_5, \neg A_6\} \quad \{\neg A_4, A_6\}$$

$$\text{OLR: } A_5 \mapsto 0 \quad \{A_3, A_4\} \quad \{A_3, \neg A_4, \neg A_6\} \quad \{\neg A_4, A_6\}$$

$$\text{DE: } A_3 \mapsto 0 \quad \{A_4\} \quad \{\neg A_4, \neg A_6\} \quad \{\neg A_4, A_6\}$$

$$\text{OLR: } A_4 \mapsto 1 \quad \{\neg A_6\} \quad \{A_6\}$$

$$\text{OLR: } A_6 \mapsto 1 \quad \{\}$$

$$D_7 := \{A_3, \neg A_4, A_5, \neg A_6\} \quad (\text{conflict clause})$$

$$D_6 := \{A_1, \neg A_2, A_3, \neg A_4, A_5\} \quad (\text{resolve } D_7, \{A_1, \neg A_2, \neg A_4, A_6\})$$

$$D_5 := \{A_1, \neg A_2, A_3, A_5\} \quad (\text{resolve } D_6, \{A_1, A_3, A_4\})$$

$$D_4 := \{A_1, \neg A_2, A_3, A_5\}$$

$$D_3 := \{A_1, \neg A_2, A_3\} \quad (\text{resolve } D_4, \{\neg A_2, \neg A_5\})$$

$$D_2 := \{A_1, \neg A_2, A_3\}$$

$$D_1 := \{\neg A_2, A_3\} \quad (\text{resolve } D_2, \{\neg A_1\})$$

Backtracking to $\{A_1 \mapsto 0, A_2 \mapsto 1\}$. Unit propagation: $A_3 \mapsto 1$.

Basic Proof Theory

Propositional Logic

(See the book by Troelstra and Schwichtenberg)

Proof rules and proof systems

Proof systems are defined by (proof or **inference**) rules of the form

$$\frac{T_1 \quad \dots \quad T_n}{T} \text{ rule-name}$$

where T_1, \dots, T_n (**premises**) and T (**conclusion**) are syntactic objects (eg formulas).

Intuitive reading: If T_1, \dots, T_n are provable, then T is provable.

Proof rules and proof systems

Proof systems are defined by (proof or **inference**) rules of the form

$$\frac{T_1 \quad \dots \quad T_n}{T} \text{ rule-name}$$

where T_1, \dots, T_n (**premises**) and T (**conclusion**) are syntactic objects (eg formulas).

Intuitive reading: If T_1, \dots, T_n are provable, then T is provable.

Degenerate case: If $n = 0$ the rule is called an **axiom** and the horizontal line is sometimes omitted.

If some U is provable, we write $\vdash U$.

Proof trees

Proofs (also: **derivations**) are drawn as trees of nested proof rules.

Example:

$$\frac{\frac{\overline{T_1} \quad \overline{U}}{\overline{T_2}} \quad \overline{T_3}}{\frac{S_1 \quad S_2}{R}}$$

We sometimes omit the names of proof rules in a proof tree if they are obvious or for space reasons. **You should always show them!**

Proof trees

Proofs (also: **derivations**) are drawn as trees of nested proof rules.

Example:

$$\frac{\frac{\overline{T_1} \quad \overline{U}}{\overline{T_2}} \quad \overline{T_3}}{\frac{S_1 \quad S_2}{R}}$$

We sometimes omit the names of proof rules in a proof tree if they are obvious or for space reasons. **You should always show them!**

Every fragment

$$\frac{T_1 \quad \dots \quad T_n}{T}$$

of a proof tree must be (an instance of) a proof rule.

All proofs must start with axioms.

The **depth** of a proof tree is the number of rules on the longest branch of the tree. Thus ≥ 1

Abbreviations

Until further notice:

\perp , \neg , \wedge , \vee , \rightarrow are primitives.

\top abbreviates $\neg\perp$

A possible simplification:

$\neg F$ abbreviates $F \rightarrow \perp$

We now consider three important proof systems:

- ▶ Sequent Calculus
- ▶ Natural Deduction
- ▶ Hilbert Systems

Sequent Calculus

Propositional Logic

Sequent Calculus

Invented by Gerhard Gentzen in 1935. Birth of proof theory.

Proof rules

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

where S_1, \dots, S_n and S are **sequents**: expressions of the form

$$\Gamma \Rightarrow \Delta$$

with Γ and Δ finite **multisets** of formulas.

Multiset = set with possibly repeated elements; using sets possible but less elegant.

Notice: \Rightarrow is just a—suggestive—separator

Intention of the calculus:

$$\Gamma \Rightarrow \Delta \text{ is provable (derivable) iff } \bigwedge \Gamma \models \bigvee \Delta \quad (\bigwedge \Gamma \rightarrow \bigvee \Delta \text{ valid})$$

Sequents: Notation

- ▶ We use set notation for multisets, e.g. $\{A, B \rightarrow C, A\}$
- ▶ Drop $\{\}$: $F_1, \dots, F_m \Rightarrow G_1, \dots, G_n$
- ▶ F, Γ abbreviates $\{F\} \cup \Gamma$ (similarly for Δ)
- ▶ Γ_1, Γ_2 abbreviates $\Gamma_1 \cup \Gamma_2$ (similarly for Δ)

Sequent Calculus rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad G, \Gamma \Rightarrow \Delta}{F \rightarrow G, \Gamma \Rightarrow \Delta} \quad \rightarrow L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \quad \vee R$$

$$\frac{F, \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \rightarrow G, \Delta} \quad \rightarrow R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

Sequent Calculus rules

Intuition: read backwards as proof search rules

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \quad \vee R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad G, \Gamma \Rightarrow \Delta}{F \rightarrow G, \Gamma \Rightarrow \Delta} \quad \rightarrow L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \quad \vee R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad G, \Gamma \Rightarrow \Delta}{F \rightarrow G, \Gamma \Rightarrow \Delta} \quad \rightarrow L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \quad \vee R$$

$$\frac{F, \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \rightarrow G, \Delta} \quad \rightarrow R$$

Sequent Calculus rules

Intuition: read backwards as proof search rules

$$\frac{}{\perp, \Gamma \Rightarrow \Delta} \quad \perp L$$

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \quad \wedge L$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \quad \vee L$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad G, \Gamma \Rightarrow \Delta}{F \rightarrow G, \Gamma \Rightarrow \Delta} \quad \rightarrow L$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \quad Ax$$

$$\frac{F, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \neg F, \Delta} \quad \neg R$$

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \wedge G, \Delta} \quad \wedge R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \quad \vee R$$

$$\frac{F, \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \rightarrow G, \Delta} \quad \rightarrow R$$

Every rule decomposes its principal formula

$$\overline{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q}$$

$$\frac{}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{F, \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \rightarrow G, \Delta} \rightarrow R$$

$$\frac{\overline{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q}}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{F, \Gamma \Rightarrow G, \Delta}{\Gamma \Rightarrow F \rightarrow G, \Delta} \rightarrow R$$

$$\frac{\overline{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \wedge L$$

$$\frac{\frac{\overline{P \vee R, Q \vee \neg R \Rightarrow P \vee Q}}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{F, G, \Gamma \Rightarrow \Delta}{F \wedge G, \Gamma \Rightarrow \Delta} \wedge L$$

$$\frac{\frac{\overline{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \vee R$$

$$\frac{\frac{\frac{P \vee R, Q \vee \neg R \Rightarrow P, Q}{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{\Gamma \Rightarrow F, G, \Delta}{\Gamma \Rightarrow F \vee G, \Delta} \vee R$$

$$\begin{array}{c}
 \frac{P \vee R, Q \vee \neg R \Rightarrow P, Q}{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R \\
 \frac{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \wedge L \\
 \Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q \rightarrow R
 \end{array}$$

$$\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \vee L$$

$$\begin{array}{c}
\frac{P, Q \vee \neg R \Rightarrow P, Q}{\frac{P \vee R, Q \vee \neg R \Rightarrow P, Q}{\frac{P \vee R, Q \vee \neg R \Rightarrow P \vee Q}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \vee R} \vee L \\
\frac{\frac{P \vee R, Q \vee \neg R \Rightarrow P \vee Q}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R \\
\\
\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \vee L
\end{array}$$

$$\frac{\frac{\frac{\overline{P, Q \vee \neg R \Rightarrow P, Q} \text{ Ax} \quad \frac{\overline{R, Q \vee \neg R \Rightarrow P, Q}}{\vee L}}{P \vee R, Q \vee \neg R \Rightarrow P, Q}}{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R}{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R$$

$$\frac{}{A, \Gamma \Rightarrow A, \Delta} \text{ Ax}$$

$$\begin{array}{c}
\frac{P, Q \vee \neg R \Rightarrow P, Q \quad Ax}{\frac{P \vee R, Q \vee \neg R \Rightarrow P, Q}{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R} \vee L \\
\frac{\frac{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \wedge L}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R \\
\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \vee L
\end{array}$$

$$\begin{array}{c}
\frac{P, Q \vee \neg R \Rightarrow P, Q \quad Ax \quad \frac{\frac{R, Q \Rightarrow P, Q \quad R, \neg R \Rightarrow P, Q}{R, Q \vee \neg R \Rightarrow P, Q} \vee L}{P \vee R, Q \vee \neg R \Rightarrow P, Q} \vee L}{\frac{P \vee R, Q \vee \neg R \Rightarrow P, Q}{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R} \wedge L \\
\frac{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q}{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R \\
\frac{F, \Gamma \Rightarrow \Delta \quad G, \Gamma \Rightarrow \Delta}{F \vee G, \Gamma \Rightarrow \Delta} \vee L
\end{array}$$

$$\begin{array}{c}
\frac{\overline{P, Q \vee \neg R \Rightarrow P, Q} \text{ Ax} \quad \frac{\overline{R, Q \Rightarrow P, Q} \text{ Ax} \quad \overline{R, \neg R \Rightarrow P, Q}}{R, Q \vee \neg R \Rightarrow P, Q} \vee L}{\overline{P \vee R, Q \vee \neg R \Rightarrow P, Q} \vee L} \vee L \\
\frac{\overline{P \vee R, Q \vee \neg R \Rightarrow P, Q}}{\overline{P \vee R, Q \vee \neg R \Rightarrow P \vee Q} \vee R} \vee R \\
\frac{\overline{(P \vee R) \wedge (Q \vee \neg R) \Rightarrow P \vee Q} \wedge L}{\overline{\Rightarrow (P \vee R) \wedge (Q \vee \neg R) \rightarrow P \vee Q} \rightarrow R} \rightarrow R
\end{array}$$

$$\overline{A, \Gamma \Rightarrow A, \Delta} \text{ Ax}$$

Proof search properties

- ▶ For every logical operator (\neg etc) there is one left and one right rule
- ▶ Every formula in the premise of a rule is a subformula of the conclusion of the rule.
This is called the **subformula property**.
 \Rightarrow no need to guess anything when applying a rule backward
- ▶ Backward rule application terminates because one operator is removed in each step.

Instances of rules

Definition

An **instance** of a rule is the result of replacing Γ and Δ by multisets of concrete formulas and F and G by concrete formulas.

Example

$$\frac{\Rightarrow P \wedge Q, A, B}{\neg(P \wedge Q) \Rightarrow A, B}$$

is an instance of

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta}$$

setting $F := P \wedge Q$, $\Gamma := \emptyset$, $\Delta := \{A, B\}$

Proof trees

Definition (Proof tree)

A **proof tree** is a tree whose nodes are sequents and where each parent-children fragment

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

is an instance of a proof rule.

(\Rightarrow all leaves must be instances of axioms)

A sequent S is **provable** (or **derivable**) if there is a proof tree with root S .

We write $\vdash_G S$ to denote that S is derivable.

Proof trees

An alternative inductive definition of proof trees:

Definition (Proof tree)

If

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

is an instance of a proof rule and

there are proof trees T_1, \dots, T_n with roots S_1, \dots, S_n

then

$$\frac{T_1 \quad \dots \quad T_n}{S}$$

is a proof tree (with root S).

What does $\Gamma \Rightarrow \Delta$ “mean”?

Definition

$$|\Gamma \Rightarrow \Delta| = \left(\bigwedge \Gamma \rightarrow \bigvee \Delta \right)$$

Example: $|\{A, B\} \Rightarrow \{P, Q\}| = (A \wedge B \rightarrow P \vee Q)$

Remember: $\bigwedge \emptyset = \top$ and $\bigvee \emptyset = \perp$

In the following slides we prove: $\vdash_G S$ iff $\models |S|$

Soundness

Lemma (Rule Equivalence)

For every rule
$$\frac{S_1 \quad \dots \quad S_n}{S}$$

- ▶ $|S| \equiv |S_1| \wedge \dots \wedge |S_n|$
- ▶ $|S|$ is a tautology iff all $|S_i|$ are tautologies

Proof: Exercise.

Theorem (Soundness of \vdash_G)

If $\vdash_G S$ then $\models |S|$.

Proof by induction on the height of the proof tree for $\vdash_G S$.

Tree must end in rule instance

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

If $n = 0$ then we vacuously have $\models |S_i|$ for all i .

If $n > 0$ then by IH we also have $\models |S_i|$ for all i .

So $\models |S_i|$ for all i , hence $\models |S|$ by Rule Equivalence.

Proof search = growing a proof tree from the root

To prove completeness we first examine the properties of the proof search procedure:

- ▶ Start from an initial sequent S_0
- ▶ At each stage we have some potentially *partial* proof tree with unproved leaves
- ▶ In each step, pick some unproved leaf S and some rule instance

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

and extend the tree with that rule instance
(creating new unproved leaves S_1, \dots, S_n)

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{}{P \vee Q \Rightarrow P \wedge Q}$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{}{P \vee Q \Rightarrow P \wedge Q}$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \vee Q \Rightarrow P}}{P \vee Q \Rightarrow P} \quad \frac{\overline{P \vee Q \Rightarrow Q}}{P \vee Q \Rightarrow Q}}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{}{P \vee Q \Rightarrow P} \vee L \quad \frac{}{P \vee Q \Rightarrow Q} \vee R}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{}{P \vee Q \Rightarrow Q} \wedge R}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \text{ Ax} \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{}{P \vee Q \Rightarrow Q} \vee R}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \quad Ax \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{}{P \vee Q \Rightarrow Q} \vee L}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \text{ Ax} \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{P \Rightarrow Q \quad \overline{Q \Rightarrow Q}}{P \vee Q \Rightarrow Q} \vee L}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \text{ Ax} \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{P \Rightarrow Q \quad \overline{Q \Rightarrow Q} \text{ Ax}}{P \vee Q \Rightarrow Q} \vee L}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Proof search terminates if ...

- ▶ there are no more unproved leaves — **success**
- ▶ there is some unproved leaf where no rule applies — **failure**
By the rules, **that leaf is of the form**

$$P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i and Q_j are atoms, no $P_i = Q_j$, and no $P_i = \perp$.

Example (failed proof)

$$\frac{\frac{\overline{P \Rightarrow P} \quad Ax \quad Q \Rightarrow P}{P \vee Q \Rightarrow P} \vee L \quad \frac{P \Rightarrow Q \quad \overline{Q \Rightarrow Q} \quad Ax}{P \vee Q \Rightarrow Q} \vee L}{P \vee Q \Rightarrow P \wedge Q} \wedge R$$

Falsifying assignments?

Proof search always terminates

Lemma (Termination)

Proof search terminates from any initial sequent S_0 .

Proof

In every step, one logical operator is removed.

- ⇒ Size of sequent decreases by 1
- ⇒ Depth of proof tree is bounded by size of S_0
- ⇒ Construction of proof tree terminates. □

Observe: Breadth only bounded by $2^{\text{size of } S_0}$.

Proof search preserves equivalence

Lemma (Search Equivalence)

At each stage of the search process,

if S_1, \dots, S_k are the unproved leaves, then $|S_0| \equiv |S_1| \wedge \dots \wedge |S_k|$

Proof by induction on the number of search steps.

Initially trivially true (base case).

When applying a rule instance

$$\frac{U_1 \quad \dots \quad U_n}{S_i}$$

we have

$$|S_0| \equiv |S_1| \wedge \dots \wedge |S_i| \wedge \dots \wedge |S_k|$$

(by IH)

$$\equiv |S_1| \wedge \dots \wedge |S_{i-1}| \wedge |U_1| \wedge \dots \wedge |U_n| \wedge |S_{i+1}| \wedge \dots \wedge |S_k|$$

(by Lemma Rule Equivalence)

Completeness

Lemma

If proof search fails, $|S_0|$ is not a tautology.

Proof If proof search fails, there is some unproved leaf

$$S = P_1, \dots, P_k \Rightarrow Q_1, \dots, Q_l$$

where all P_i, Q_j atoms, no $P_i = Q_j$ and no $P_i = \perp$.

Any assignment \mathcal{A} with $\mathcal{A}(P_i) = 1$ (for all i)

and $\mathcal{A}(Q_j) = 0$ (for all j) satisfies $\mathcal{A}(|S|) = 0$.

Thus $\mathcal{A}(|S_0|) = 0$ by Lemma Search Equivalence. □

Because of soundness of \vdash_G :

Corollary

Starting with some fixed S_0 , proof search cannot both fail (for some choices) and succeed (for other choices).

\Rightarrow no need for backtracking upon failure!

Completeness

Theorem (Completeness)

If $\models S$ then $\vdash_G S$.

Proof by contraposition: if not $\vdash_G S$ then proof search must fail.
Therefore $\not\models S$.

Corollary

Proof search is a decision procedure: it always terminates and it succeeds iff $\models S$.

Multisets versus sets

Termination only because of multisets.

With sets, the principal formula may get duplicated:

$$\frac{\Gamma \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta} \quad \neg L \quad \Gamma := \{\neg F\} \rightsquigarrow \frac{\neg F \Rightarrow F, \Delta}{\neg F \Rightarrow \Delta}$$

An alternative formulation of the set version:

$$\frac{\Gamma \setminus \{\neg F\} \Rightarrow F, \Delta}{\neg F, \Gamma \Rightarrow \Delta}$$

Gentzen used sequences (hence “sequent calculus”)

Admissible Rules and Cut Elimination

Admissible rules

Definition

A rule

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

is **admissible** if $\vdash_G S_1, \dots, \vdash_G S_n$ together imply $\vdash_G S$.

\Rightarrow Admissible rules can be used in proofs like normal rules

Admissibility of

$$\frac{S_1 \quad \dots \quad S_n}{S}$$

can be shown semantically (using \vdash_G iff \models)

by proving that $\models |S_1|, \dots, \models |S_n|$ together imply $\models |S|$.

Proof theory is interested in **syntactic proofs** that show **how** to eliminate admissible rules.

Cut elimination rule

Theorem (Gentzen's *Hauptsatz*)

The cut elimination rule

$$\frac{\Gamma \Rightarrow F, \Delta \quad \Gamma, F \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \quad \text{cut}$$

is admissible.

Proof Omitted.

Proofs with cut elimination can be much shorter than proofs without!

But: applying the rule needs creativity! (find the right F)

Intuitively: Proof of Gentzen's theorem shows how to replace creativity by calculation.

Many applications.

Natural Deduction

Propositional Logic

(See the book by Troelstra and Schwichtenberg)

Natural deduction (Gentzen 1935) aims at *natural* proofs.

It formalizes good mathematical practice.

Resolution, but also sequent calculus, aim at proof search.

Main principles: Introduction and elimination rules

1. For every logical operator \oplus there are two kinds of rules:

▶ **Introduction rules:** How to prove $F \oplus G$

$$\frac{\dots}{F \oplus G}$$

▶ **Elimination rules:** What can be proved from $F \oplus G$

$$\frac{F \oplus G \quad \dots}{\dots}$$

Examples

$$\frac{A \quad B}{A \wedge B} \quad \wedge I \qquad \frac{F \wedge G}{F} \quad \wedge E_1 \qquad \frac{F \wedge G}{G} \quad \wedge E_2$$

Main principles: Local assumptions

2. Proof can contain subproofs with *local/closed* assumptions

Example

Inference rule formalizing “if from the local assumption F we can prove G then we can prove $F \rightarrow G$ ”:

$$\frac{\begin{array}{c} [F] \\ \vdots \\ G \end{array}}{F \rightarrow G} \rightarrow I$$

A proof tree:

$$\frac{\frac{\begin{array}{c} [P] \quad Q \\ \hline P \wedge Q \end{array}}{P \rightarrow P \wedge Q} \rightarrow I}{\quad} \wedge I$$

“From the (open) assumption Q we can prove $P \rightarrow P \wedge Q$.”

In symbols: $Q \vdash_N P \rightarrow P \wedge Q$

Main principles: Growing the proof tree

Upwards:

Main principles: Growing the proof tree

Upwards:

$$\overline{P \rightarrow P \wedge Q}$$

Main principles: Growing the proof tree

Upwards:

$$\overline{P \rightarrow P \wedge Q} \quad \rightarrow I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\overline{P \wedge Q}}{P \rightarrow P \wedge Q} \rightarrow I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\overline{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \begin{matrix} \wedge I \\ \rightarrow I \end{matrix}$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{P \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \begin{array}{l} \wedge I \\ \rightarrow I \end{array}$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \begin{array}{l} \wedge I \\ \rightarrow I \end{array}$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \quad \rightarrow I$$

Downwards:

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \quad \rightarrow I$$

Downwards:

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \quad \rightarrow I$$

Downwards:

$$\frac{P \quad Q}{\quad}$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

Downwards:

$$\frac{P \quad Q}{\quad} \quad \wedge I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

Downwards:

$$\frac{P \quad Q}{P \wedge Q} \quad \wedge I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

Downwards:

$$\frac{P \quad Q}{P \wedge Q} \quad \wedge I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

Downwards:

$$\frac{P \quad Q}{\frac{P \wedge Q}{P \rightarrow P \wedge Q}} \quad \wedge I \rightarrow I$$

Main principles: Growing the proof tree

Upwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

Downwards:

$$\frac{\frac{[P] \quad Q}{P \wedge Q}}{P \rightarrow P \wedge Q} \quad \wedge I \rightarrow I$$

ND proof trees

- ▶ The nodes of a ND proof tree are (labeled by) formulas.
- ▶ Leaf nodes are called **assumptions**.
- ▶ The root is called the **conclusion**.

- ▶ Assumptions can be **open** or **closed**.
- ▶ Closed assumptions are written **[F]**.
- ▶ $\Gamma \vdash_N F$ denotes that there is a proof tree with conclusion F whose open assumptions belong to the set of formulas Γ .
(Reading: F is provable (derivable) from Γ .)

Intuition:

- ▶ A proof tree shows that the **conjunction** of the **open** assumptions entails the conclusion.
- ▶ Closed assumptions are **auxiliary local** assumptions in a subproof that have been closed (“discharged”) by some proof rule like $\rightarrow I$.

ND proof trees

ND proof trees are defined inductively:

- ▶ Every formula F is a ND proof tree with open assumption F and conclusion F .
(Intuition: From F we can prove F .)
- ▶ Larger proof trees are constructed using the rules of ND:
 - Introduction and Elimination rules for $\wedge, \vee, \rightarrow, \neg$, plus
 - a rule for \perp .

The application of a rule (backwards or forwards) adds new nodes to the tree and possibly closes some assumptions:

Natural Deduction rules

$$\frac{F \quad G}{F \wedge G} \wedge I$$

$$\frac{F \wedge G}{F} \wedge E_1 \quad \frac{F \wedge G}{G} \wedge E_2$$

$$\frac{\begin{array}{c} [F] \\ \vdots \\ G \end{array}}{F \rightarrow G} \rightarrow I$$

$$\frac{F \rightarrow G \quad F}{G} \rightarrow E$$

$$\frac{F}{F \vee G} \vee I_1 \quad \frac{G}{F \vee G} \vee I_2$$

$$\frac{F \vee G \quad \begin{array}{c} [F] \\ \vdots \\ H \end{array} \quad \begin{array}{c} [G] \\ \vdots \\ H \end{array}}{H} \vee E$$

$$\frac{\begin{array}{c} [F] \\ \vdots \\ \perp \end{array}}{\neg F} \neg I$$

$$\frac{\neg F \quad F}{\perp} \neg E$$

$$\frac{\begin{array}{c} [\neg F] \\ \vdots \\ \perp \end{array}}{F} \perp$$

Natural Deduction rules

How to read a rule

$$\frac{\dots \quad \begin{array}{c} [F] \\ \vdots \\ G \end{array} \quad \dots}{\dots} \quad r$$

Forward:

When applying rule r , in the proof of G we can close all (or some) of the assumptions F .

Backward:

In the subproof of G we can use the local assumption $[F]$.

We can use labels to show which rule application closed which assumptions (the slides won't but you must!).

Examples of proofs

$$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P:$$

Examples of proofs

$$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P:$$

$$\overline{\neg Q \rightarrow \neg P}$$

Examples of proofs

$$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P:$$

$$\overline{\neg Q \rightarrow \neg P} \rightarrow I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\overline{\neg P}}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\overline{\neg P} \quad \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{\perp}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{\frac{\perp}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1}{\quad} \neg E:3$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{\frac{\perp}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1}{Q} [\neg Q]^1 \neg E:3$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{\frac{\perp}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1}{\frac{Q \quad [\neg Q]^1}{\perp} \rightarrow E:4}{\neg Q \rightarrow \neg P} \neg E:3}$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

$$\neg P \wedge \neg Q$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{}{\neg P \wedge \neg Q} \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

$$\frac{\overline{\neg P} \quad \overline{\neg Q}}{\neg P \wedge \neg Q} \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\overline{\neg P} \quad \overline{\neg Q}}{\neg P \wedge \neg Q} \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{\perp}{\neg P} \neg I:2 \quad \overline{\neg Q}}{\neg P \wedge \neg Q} \wedge I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{\perp}{\neg P} \neg I:2 \quad \overline{\neg Q}}{\neg P \wedge \neg Q} \wedge I:1}{\perp} \neg E:3$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{\overline{P \vee Q} \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \wedge I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{\overline{P \vee Q} \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \wedge I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \wedge I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \quad \frac{}{\neg Q} \neg I:5}{\neg P \wedge \neg Q} \wedge I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \quad \frac{\perp}{\neg Q} \neg I:5}{\neg P \wedge \neg Q} \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\frac{\perp}{\neg P} \neg I:2} \neg E:6 \quad \frac{\perp}{\neg Q} \neg I:5 \vee I:1}{\neg P \wedge \neg Q}$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg P \wedge \neg Q} \wedge I:1 \quad \frac{\overline{P \vee Q} \quad \neg(P \vee Q)}{\perp} \neg E:6}{\neg Q} \neg I:5 \quad \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg I:2 \quad \neg E:3}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg I:2 \quad \neg E:3 \quad \frac{\overline{P \vee Q} \vee I:7 \quad \neg(P \vee Q)}{\perp} \neg I:5 \quad \neg E:6}{\neg P \wedge \neg Q} \vee I:1$$

Examples of proofs

$P \rightarrow Q \vdash_N \neg Q \rightarrow \neg P$:

$$\frac{\frac{[P]^2 \quad P \rightarrow Q}{Q} \rightarrow E:4 \quad [\neg Q]^1}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg Q \rightarrow \neg P} \rightarrow I:1$$

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

$$\frac{\frac{[P]^2}{P \vee Q} \vee I:4 \quad \neg(P \vee Q)}{\perp} \neg E:3}{\neg P} \neg I:2}{\neg P \wedge \neg Q} \wedge I:1 \quad \frac{\frac{[Q]^5}{P \vee Q} \vee I:7 \quad \neg(P \vee Q)}{\perp} \neg E:6}{\neg Q} \neg I:5 \quad \vee I:1$$

Soundness

Lemma (Soundness)

If $\Gamma \vdash_N F$ then $\Gamma \models F$

Proof by induction on the depth of the proof tree for $\Gamma \vdash_N F$.

Base: The tree has only one node F and $F \in \Gamma$.

Step: Case analysis of first rule applied (upwards).

Case: first rule is $\frac{G \rightarrow F \quad G}{F} \rightarrow E$

Let \mathcal{A} arbitrary such that $\mathcal{A}(\Gamma) = 1$. We prove $\mathcal{A}(F) = 1$

IH: $\Gamma \models G \rightarrow F$ and $\Gamma \models G$

IH and $\mathcal{A}(\Gamma) = 1$ yields $\mathcal{A}(G \rightarrow F) = 1$ and $\mathcal{A}(G) = 1$.

So $\mathcal{A}(F) = 1$

Soundness

Case: first rule is

$$\frac{\begin{array}{c} [G] \\ \vdots \\ F \end{array}}{G \rightarrow F} \rightarrow I$$

To show: $\Gamma \models G \rightarrow F$

IH: $\Gamma, G \models F$

$\Gamma \models G \rightarrow F$

iff for all \mathcal{A} : $\mathcal{A} \models \Gamma \Rightarrow \mathcal{A} \models G \rightarrow F$

iff for all \mathcal{A} : $\mathcal{A} \models \Gamma \Rightarrow (\mathcal{A} \models G \Rightarrow \mathcal{A} \models F)$

iff for all \mathcal{A} : $(\mathcal{A} \models \Gamma \text{ and } \mathcal{A} \models G) \Rightarrow \mathcal{A} \models F$

iff IH

Towards completeness: ND can simulate truth tables

Lemma (Tertium non datur)

$\vdash_N F \vee \neg F$

Proof:

$$\frac{\frac{\frac{[\neg(F \vee \neg F)]^1}{\perp} \quad \frac{[\neg F]^4}{F \vee \neg F}}{\perp} \quad \vee I_2:6}{\perp} \quad \neg E:5}{\frac{\perp}{F} \quad \perp:4}{F \vee \neg F} \quad \vee I_1:3}{[\neg(F \vee \neg F)]^1 \quad \neg E:2}{\frac{\perp}{F \vee \neg F} \quad \perp:1}$$

Towards completeness: ND can simulate truth tables

Definition

$$F^{\mathcal{A}} := \begin{cases} F & \text{if } \mathcal{A}(F) = 1 \\ \neg F & \text{if } \mathcal{A}(F) = 0 \end{cases}$$

Lemma (1)

If $\text{atoms}(F) \subseteq \{A_1, \dots, A_n\}$ then $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N F^{\mathcal{A}}$ for every \mathcal{A} .

Proof By induction on F .

Only the case $F = G \rightarrow H$. Three subcases:

$\mathcal{A}(H) = 1$. Then $H^{\mathcal{A}} = H$, $(G \rightarrow H)^{\mathcal{A}} = G \rightarrow H$.

To prove: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N G \rightarrow H$. By IH: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N H$.

$$\frac{\overline{H} \quad IH}{G \rightarrow H} \rightarrow I$$

Towards completeness: ND can simulate truth tables

$\mathcal{A}(G) = 0$. Then $G^{\mathcal{A}} = \neg G$, $(G \rightarrow H)^{\mathcal{A}} = G \rightarrow H$.

To prove: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N G \rightarrow H$. By IH: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N \neg G$.

$$\frac{[G] \quad \overline{\neg G} \quad \text{IH}}{\perp} \neg E$$

$$\frac{\perp \quad \perp}{H} \perp$$

$$\frac{H}{G \rightarrow H} \rightarrow I$$

$\mathcal{A}(G) = 1$ and $\mathcal{A}(H) = 0$. Then $G^{\mathcal{A}} = G$, $H^{\mathcal{A}} = \neg H$,
 $(G \rightarrow H)^{\mathcal{A}} = \neg(G \rightarrow H)$.

To prove: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N \neg(G \rightarrow H)$.

By IH: $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N G$ and $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N \neg H$.

$$\frac{[G \rightarrow H] \quad \overline{G} \quad \text{IH}}{H} \rightarrow E$$

$$\frac{H \quad \overline{\neg H} \quad \text{IH}}{\perp} \perp$$

$$\frac{\perp}{\neg(G \rightarrow H)} \rightarrow I$$

Towards completeness: ND can simulate truth tables

Corollary (Cases)

If $F, \Gamma \vdash_N G$ and $\neg F, \Gamma \vdash_N G$ then $\Gamma \vdash_N G$.

Proof: By Lemma (Tertium non datur) $F, \Gamma \vdash_N F \vee \neg F$.

Apply

$$\frac{F \vee \neg F \quad \begin{array}{c} [F] \\ \vdots \\ G \end{array} \quad \begin{array}{c} [\neg F] \\ \vdots \\ G \end{array}}{G} \vee E$$

Completeness

Lemma (2)

If $\text{atoms}(F) = \{A_1, \dots, A_n\}$ and $\models F$ then $A_1^{\mathcal{A}}, \dots, A_k^{\mathcal{A}} \vdash_N F$ for every \mathcal{A} and for all $k \leq n$.

Proof by (downward) induction on $k = n, \dots, 0$.

$k = n$. $A_1^{\mathcal{A}}, \dots, A_n^{\mathcal{A}} \vdash_N F$ holds by Lemma (1) and $F^{\mathcal{A}} = F$ because F is valid.

$k < n$. By IH $A_1^{\mathcal{A}}, \dots, A_k^{\mathcal{A}} \vdash_N F$ for every \mathcal{A} .

To prove: $A_1^{\mathcal{A}}, \dots, A_{k-1}^{\mathcal{A}} \vdash_N F$ for every \mathcal{A} .

Let \mathcal{A} arbitrary. Define $\bar{\mathcal{A}}$ by $\bar{\mathcal{A}}(A_i) = \mathcal{A}(A_i)$ for every $i < k$ and $\bar{\mathcal{A}}(A_k) = 1 - \mathcal{A}(A_k)$.

Assume w.l.o.g. $\mathcal{A}(A_k) = 1$ (otherwise swap \mathcal{A} and $\bar{\mathcal{A}}$).

Then $A_i^{\bar{\mathcal{A}}} = A_i^{\mathcal{A}}$ for every $i < k$, and $A_k^{\bar{\mathcal{A}}} = \neg A_k^{\mathcal{A}}$.

Taking $F := \mathcal{A}_k^{\mathcal{A}}$, $\Gamma := A_1^{\mathcal{A}}, \dots, A_{k-1}^{\mathcal{A}}$, and $G := F$ in Corollary (Cases) yields $A_1^{\mathcal{A}}, \dots, A_{k-1}^{\mathcal{A}} \vdash_N F$.

Completeness

Theorem (Completeness)

If $\Gamma \models F$ then $\Gamma \vdash_N F$.

Proof Only for $\Gamma := G$ (general case left as exercise).

Assume $G \models F$.

We have $\models G \rightarrow F$ and so $\vdash_N G \rightarrow F$ by Lemma (2) with $n = 0$.

Applying

$$\frac{G \rightarrow F \quad G}{F} \rightarrow E$$

yields $G \vdash_N F$.

Hilbert Systems

Propositional Logic

(See the book by Troelstra and Schwichtenberg)

Easy to define, hard to use.
No context management.

Hilbert systems and proof trees

A **Hilbert system** for propositional logic consists of

- ▶ a set of **axioms** (formulas over formula variables)
- ▶ and a single inference rule, $\rightarrow E$ or **modus ponens**:

$$\frac{F \rightarrow G \quad F}{G} \rightarrow E$$

Nodes of a proof tree are (labeled with) formulas.

Proof trees are defined inductively:

- ▶ Every formula F is a proof tree.
- ▶ Larger proof trees are constructed using $\rightarrow E$ (and $\rightarrow E$ only).

$\Gamma \vdash_H F$ denotes that there is a proof tree with root F whose leaves are either instances of axioms or elements of Γ (assumptions).

Alternative presentation

Proofs in Hilbert systems are frequently shown as lists of lines

$$\begin{array}{l} 1: F_1 \quad \textit{justification}_1 \\ 2: F_2 \quad \textit{justification}_2 \\ \vdots \\ i: F_i \quad \textit{justification}_i \\ \vdots \\ n: F_n \quad \textit{justification}_n \end{array}$$

where $\textit{justification}_i$ is either $\left\{ \begin{array}{l} \textit{assumption}, \\ \textit{axiom}, \\ \rightarrow E (j, k), \text{ with } j, k < i. \end{array} \right.$

Notational convention:

$$F \rightarrow G \rightarrow H \text{ means } F \rightarrow (G \rightarrow H)$$

Note: $F \rightarrow G \rightarrow H \equiv F \wedge G \rightarrow H$
 $F \rightarrow G \rightarrow H \not\equiv (F \rightarrow G) \rightarrow H$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 :

2 :

3 :

4 :

5 : $A \rightarrow A$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 :

2 :

3 : _____ $\rightarrow A \rightarrow A$

4 : _____

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 :

2 :

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$

4 : $A \rightarrow A \rightarrow A$

5 : $A \rightarrow A$

$\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 :

2 :

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$

4 : $A \rightarrow A \rightarrow A$

A1

5 : $A \rightarrow A$

$\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1: _____ $\rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$

2: _____

3: $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4: $A \rightarrow A \rightarrow A$ A1

5: $A \rightarrow A$ $\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 : $(A \rightarrow (A \rightarrow A) \rightarrow A) \rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$

2 : $A \rightarrow (A \rightarrow A) \rightarrow A$

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4 : $A \rightarrow A \rightarrow A$ A1

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 : $(A \rightarrow (A \rightarrow A) \rightarrow A) \rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$

2 : $A \rightarrow (A \rightarrow A) \rightarrow A$ A1

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4 : $A \rightarrow A \rightarrow A$ A1

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 : $(A \rightarrow (A \rightarrow A) \rightarrow A) \rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ A2

2 : $A \rightarrow (A \rightarrow A) \rightarrow A$ A1

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4 : $A \rightarrow A \rightarrow A$ A1

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 : $(A \rightarrow (A \rightarrow A) \rightarrow A) \rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ A2

2 : $A \rightarrow (A \rightarrow A) \rightarrow A$ A1

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4 : $A \rightarrow A \rightarrow A$ A1

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

$\Rightarrow \vdash_H A \rightarrow A$

A simple Hilbert system

Axioms: $F \rightarrow G \rightarrow F$ (A1)

$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H$ (A2)

A proof of $A \rightarrow A$:

1 : $(A \rightarrow (A \rightarrow A) \rightarrow A) \rightarrow (A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ A2

2 : $A \rightarrow (A \rightarrow A) \rightarrow A$ A1

3 : $(A \rightarrow A \rightarrow A) \rightarrow A \rightarrow A$ $\rightarrow E : 2, 1$

4 : $A \rightarrow A \rightarrow A$ A1

5 : $A \rightarrow A$ $\rightarrow E : 3, 4$

$\Rightarrow \vdash_H A \rightarrow A$

Observe: The same proof can be used to derive $F \rightarrow F$ for any formula F .

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof “ \Leftarrow ”: Assume $\Gamma \vdash_H F \rightarrow G$. We prove $F, \Gamma \vdash_H G$.

$$\begin{aligned} & \Gamma \vdash_H F \rightarrow G \\ \Rightarrow & F, \Gamma \vdash_H F \rightarrow G \\ \Rightarrow & F, \Gamma \vdash_H G \quad \text{by } \rightarrow E \text{ because } F, \Gamma \vdash_H F \end{aligned}$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof " \Rightarrow ": Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Base: $n = 1$. Then either $G \in \Gamma \cup \{F\}$ or G is instance of axiom.

- ▶ $G = F$. To prove: $\Gamma \vdash_H F \rightarrow F$. Done earlier.
- ▶ $G \in \Gamma$ or instance of axiom.

$$\begin{array}{ll} 1 : G & \\ 2 : G \rightarrow F \rightarrow G & \text{A1} \\ 3 : F \rightarrow G & \rightarrow E : 1, 2 \end{array}$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof " \Rightarrow ": Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Step: $n > 1$. Assume last $\rightarrow E$ gives G from $H \rightarrow G$ and H .

IH: $\Gamma \vdash_H F \rightarrow H$ and $\Gamma \vdash_H F \rightarrow H \rightarrow G$.

To prove: $\Gamma \vdash_H F \rightarrow G$.

$$F \rightarrow G$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof " \Rightarrow ": Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Step: $n > 1$. Assume last $\rightarrow E$ gives G from $H \rightarrow G$ and H .

IH: $\Gamma \vdash_H F \rightarrow H$ and $\Gamma \vdash_H F \rightarrow H \rightarrow G$.

To prove: $\Gamma \vdash_H F \rightarrow G$.

$$F \rightarrow G$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof “ \Rightarrow ”: Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Step: $n > 1$. Assume last $\rightarrow E$ gives G from $H \rightarrow G$ and H .

IH: $\Gamma \vdash_H F \rightarrow H$ and $\Gamma \vdash_H F \rightarrow H \rightarrow G$.

To prove: $\Gamma \vdash_H F \rightarrow G$.

$$\frac{(F \rightarrow H) \rightarrow F \rightarrow G \quad F \rightarrow H}{F \rightarrow G}$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof " \Rightarrow ": Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Step: $n > 1$. Assume last $\rightarrow E$ gives G from $H \rightarrow G$ and H .

IH: $\Gamma \vdash_H F \rightarrow H$ and $\Gamma \vdash_H F \rightarrow H \rightarrow G$.

To prove: $\Gamma \vdash_H F \rightarrow G$.

$$\frac{\frac{(F \rightarrow H \rightarrow G) \rightarrow (F \rightarrow H) \rightarrow F \rightarrow G \quad F \rightarrow H \rightarrow G}{(F \rightarrow H) \rightarrow F \rightarrow G} \quad F \rightarrow H}{F \rightarrow G}$$

Theorem (Deduction Theorem)

In any Hilbert-system that contains the axioms A1 and A2:

$$F, \Gamma \vdash_H G \quad \text{iff} \quad \Gamma \vdash_H F \rightarrow G$$

Proof " \Rightarrow ": Assume $F, \Gamma \vdash_H G$ with a proof of length n .
We prove $\Gamma \vdash_H F \rightarrow G$ by induction on n .

Step: $n > 1$. Assume last $\rightarrow E$ gives G from $H \rightarrow G$ and H .

IH: $\Gamma \vdash_H F \rightarrow H$ and $\Gamma \vdash_H F \rightarrow H \rightarrow G$.

To prove: $\Gamma \vdash_H F \rightarrow G$.

$$\frac{\text{A2: } (F \rightarrow H \rightarrow G) \rightarrow (F \rightarrow H) \rightarrow F \rightarrow G \quad F \rightarrow H \rightarrow G}{\frac{(F \rightarrow H) \rightarrow F \rightarrow G}{F \rightarrow G} \quad F \rightarrow H}$$

Hilbert System

From now on \vdash_H refers to the following set of axioms:

$$F \rightarrow G \rightarrow F \quad (\text{A1})$$

$$(F \rightarrow G \rightarrow H) \rightarrow (F \rightarrow G) \rightarrow F \rightarrow H \quad (\text{A2})$$

$$F \rightarrow G \rightarrow F \wedge G \quad (\text{A3})$$

$$F \wedge G \rightarrow F \quad (\text{A4})$$

$$F \wedge G \rightarrow G \quad (\text{A5})$$

$$F \rightarrow F \vee G \quad (\text{A6})$$

$$G \rightarrow F \vee G \quad (\text{A7})$$

$$F \vee G \rightarrow (F \rightarrow H) \rightarrow (G \rightarrow H) \rightarrow H \quad (\text{A8})$$

$$(\neg F \rightarrow \perp) \rightarrow F \quad (\text{A9})$$

We prove soundness and completeness.

Relating Hilbert and Natural Deduction

Theorem (Hilbert can simulate ND)

If $\Gamma \vdash_N F$ then $\Gamma \vdash_H F$

Proof. Translation in two steps:

$$\vdash_N \overset{(1)}{\rightsquigarrow} \vdash_H + \rightarrow I \overset{(2)}{\rightsquigarrow} \vdash_H$$

1. Transform a ND-proof tree into a proof tree containing Hilbert axioms, $\rightarrow E$, and $\rightarrow I$ by replacing all other ND rules by Hilbert proofs with $\rightarrow I$

Principle: ND rule \rightsquigarrow 1 axiom + $\rightarrow I/E$

2. Eliminate $\rightarrow I$ rules using the Deduction Theorem

Theorem (ND can simulate Hilbert)

If $\Gamma \vdash_H F$ then $\Gamma \vdash_N F$

Proof by induction on the length of the Hilbert proof of F .

- ▶ Every Hilbert axiom is provable in ND (Exercise!).
- ▶ $\rightarrow E$ is also available in ND.

Corollary

$\Gamma \vdash_H F$ iff $\Gamma \vdash_N F$

Corollary (Soundness and completeness)

$\Gamma \vdash_H F$ iff $\Gamma \models F$