



Reasoning w/ propositional logic

Remember what we've developed so far

- Logical sentences
- And, or, not, implies (entailment), iff (equivalence)
- Syntax vs. semantics
- Truth tables
- Satisfiability, proof by contradiction



Logical Equivalences

Know these equivalences

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

Figure 7.11 Standard logical equivalences. The symbols α , β , and γ stand for arbitrary sentences of propositional logic.



Reasoning with propositional logic

Inference Rules

$$\frac{\alpha \Rightarrow \beta, \quad \alpha}{\beta}$$

- Modus Ponens:
 - Whenever sentences of form $\alpha \Rightarrow \beta$ and α are given the sentence β can be inferred
 - R_1 : Green $\Rightarrow$ Martian
 - R_2 : Green
 - Inferred: Martian

Reasoning with propositional logic



Inference Rules

- And-Elimination
 - Any of conjuncts can be inferred
 - R_1 : Martian $\wedge$ Green
 - Inferred: Martian
 - Inferred: Green

$$\frac{\alpha \wedge \beta}{\alpha}$$

Use truth tables if you want to confirm inference rules

Inference Rules



Name	Rule of Inference	Tautology
Modus [ponendo] ponens = “the way that affirms by affirming”	$\begin{array}{l} p \\ p \rightarrow q \\ \hline \therefore q \end{array}$	$(p \wedge (p \rightarrow q)) \rightarrow q$
Modus [tollendo] tollens = “the way that denies by denying”	$\begin{array}{l} \neg q \\ p \rightarrow q \\ \hline \therefore \neg p \end{array}$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$
Transitivity (Hypothetical syllogism)	$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
Elimination (Disjunctive syllogism)	$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$	$((p \vee q) \wedge \neg p) \rightarrow q$
Generalization	$\begin{array}{l} p \\ \hline \therefore p \vee q \end{array}$	$p \rightarrow (p \vee q)$
Simplification	$\begin{array}{l} p \wedge q \\ \hline \therefore p \end{array}$	$(p \wedge q) \rightarrow p$
Conjunction	$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$
Resolution	$\begin{array}{l} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$

Example of a proof



If today is Tuesday, I have a test in Mathematics or Economics. If my Economics professor is sick, I will not have a test in Economics. Today is Tuesday, and my Economics professor is sick. Therefore, I will have a test in Mathematics.

Converting to logical notation:

t = today is Tuesday

m = I have a test in Mathematics

e = I have a test in Economics

s = My Economics Professor is sick.

So the argument is:

$t \rightarrow (m \vee e)$

$s \rightarrow \neg e$

$t \wedge s$

$\therefore m$

- | | |
|-------------------------------|-------------------------------------|
| 1. $t \wedge s$ | premise |
| 2. t | from (1) by simplification Law |
| 3. $t \rightarrow (m \vee e)$ | premise |
| 4. $m \vee e$ | from (2) and (3) by modus ponens |
| 5. s | from (1) by simplification Law |
| 6. $s \rightarrow \neg e$ | premise |
| 7. $\neg e$ | from (5) and (6) by modus ponens |
| 8. $e \vee m$ | from (4) by commutative law |
| 9. m | from (7) and (8) by elimination Law |

Constructing a proof



*Proving **is like** searching*

- Find sequence of logical inference rules that lead to desired result
- Note the explosion of propositions
 - Good proof methods ignore the countless irrelevant propositions



How many inferences?

Previous example relied on application of inference rules to generate new sentences

- When have you drawn enough inferences to prove something?
 - Too many make search process take longer
 - Too few halt logical progression and make proof process **incomplete**

Completeness Theorem



Definition:

Formula α is a **tautology** if for every valuation v (in every model v), $v(\alpha) = 1$ (true)

Axioms

A1: $A \rightarrow (B \rightarrow A)$

A2: $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

A3: $(\neg B \rightarrow \neg A) \rightarrow [(\neg B \rightarrow A) \rightarrow B]$

Rule of Inference:

R: $[A, A \rightarrow B] / B$

Definition:

Formula α is a theorem if it can be proved from Axioms A1, A2, A3 using rule R.

Proposition 1. If formula α is a theorem, then α is a tautology (soundness)

Proposition 2. If formula α is a tautology, then α is a theorem (completeness)

Proposition 3. $\vdash \alpha$ iff $\models \alpha$; $F \vdash \alpha$ iff $F \models \alpha$, where F –set of formulas.



What about “and” clauses?

Resolution only applies to “or” clauses

- Every sentence of propositional logic can be transformed to a logically equivalent conjunction of disjunctions of literals

Conjunctive Normal Form (CNF)

- A sentence expressed as conjunction of disjunction of literals
 - k-CNF: exactly k literals per clause

$$(\ell_{1,1} \vee \dots \vee \ell_{1,k}) \wedge \dots \wedge (\ell_{n,1} \vee \dots \vee \ell_{n,k})$$



Converting formula to CNF

The steps are as follows:

1. Eliminate $\leftrightarrow$ replacing $\alpha \leftrightarrow \beta$ by $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
2. Eliminate $\rightarrow$ by replacing $\alpha \rightarrow \beta$ by $\neg \alpha \vee \beta$
3. CNF requires $\neg$ to appear only in literals so we move $\neg$ inwards by repeated application of the following equivalence forms:

$\neg(\neg \alpha) = \alpha$ (double negation elimination)

$\neg(\alpha \wedge \beta) = (\neg \alpha \vee \neg \beta)$ (de Morgan)

$\neg(\alpha \vee \beta) = (\neg \alpha \wedge \neg \beta)$ (de Morgan)

4. Now, we apply the distributivity law

$(\alpha \wedge \beta) \vee \gamma = (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$

by distributing $\vee$ over $\wedge$ wherever possible.

$(A \rightarrow (B \vee C)) \wedge (A \leftarrow (B \vee C))$

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

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

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

Example: Convert the formula $A \leftrightarrow (B \vee C)$ to CNF



An algorithm for resolution

We wish to prove KB entails α

- Must show $(KB \wedge \sim\alpha)$ is unsatisfiable
 - No possible way for KB to entail (not α)
 - Proof by contradiction

Proof by Contradiction - Example



Proof attributed to Euclid (300 BC)

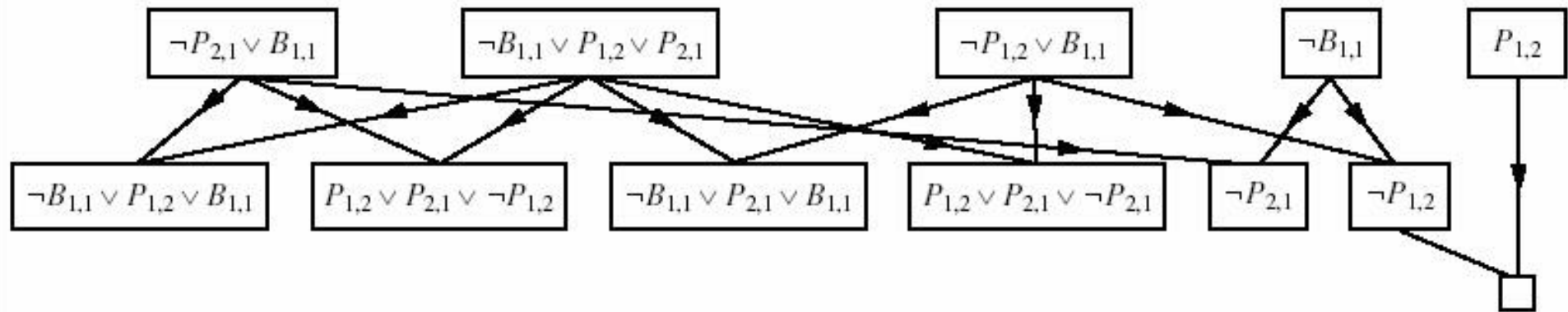


Theorem. There are infinitely many prime numbers.

Proof. Assume to the contrary that there are only finitely many prime numbers, and all of them are listed as follows: $p_1, p_2, \dots, p_n$. Consider the number $q = p_1 p_2 \dots p_n + 1$. $\square$ The number q is either prime or composite. If we divided any of the listed primes p_i into q , there would result a remainder of 1 for each $i = 1, 2, \dots, n$. Thus, q cannot be composite. We conclude that q is a prime number, not among the primes listed above, contradicting our assumption that **all** primes are in the list $p_1, p_2, \dots, p_n$.



Example of resolution in KB



Partial application of PL-RESOLUTION to a simple inference
 $\neg P_{1,2}$ is shown to follow from the first four clauses in the top row.

Proof that there is not a pit in $P_{1,2}$: $\sim P_{1,2}$

- $KB \wedge P_{1,2}$ leads to empty clause (resolution refutation)
- Therefore $\sim P_{1,2}$ is true



Resolution Refutation

Problem 1. Show by resolution refutation that each of the following formulas is a tautology:

- (a) $(P \rightarrow Q) \rightarrow [(R \vee P) \rightarrow (R \vee Q)]$
- (b) $[(P \rightarrow Q) \rightarrow P] \rightarrow P$
- (c) $(\neg P \rightarrow P) \rightarrow P$
- (d) $(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$

Problem 2. /language alphabet – $\{P, Q, R\}$ /

Let's consider interpretation v where $v(P) = F$, $v(Q) = T$, $v(R) = T$. Does v satisfy the following propositional formulas?

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



Horn Clauses

Horn Clause

- Disjunction of literals with at most one is positive
 - $(\sim a \vee \sim b \vee \sim c \vee d)$
 - $(\sim a \vee b \vee c \vee \sim d)$ **Not a Horn Clause**



Horn Clauses

Can be written as a special implication

- $(\sim a \vee \sim b \vee c) \Leftrightarrow (a \wedge b) \Rightarrow c$
 - $(\sim a \vee \sim b \vee c) \equiv (\sim(a \wedge b) \vee c)$... de Morgan
 - $(\sim(a \wedge b) \vee c) \equiv ((a \wedge b) \Rightarrow c)$... implication elimination



Horn Clauses

Permit straightforward inference determination

- Forward chaining (repeated application of modus ponens)
- Backward chaining (an inference method described as working backward from the goal).

Example of backward chaining: RS-strategy (Project 1)



Forward Chaining

Properties

- Sound
- Complete
 - All entailed atomic sentence will be derived

Data Driven

- Start with what we know
- Derive new info until we discover what we want