What is a logic?

- A formal language

What is a logic?

- A formal language

What is a logic?

- A formal language
  - Syntax – what expressions are legal

What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean

What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean
  - Proof system – a way of manipulating syntactic expressions to get other syntactic expressions (which will tell us something new)

What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean
  - Proof system – a way of manipulating syntactic expressions to get other syntactic expressions (which will tell us something new)
- Why proofs? Two kinds of inferences an agent might want to make:

What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean
  - Proof system – a way of manipulating syntactic expressions to get other syntactic expressions (which will tell us something new)
- Why proofs? Two kinds of inferences an agent might want to make:
  - Multiple percepts => conclusions about the world

What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean
  - Proof system – a way of manipulating syntactic expressions to get other syntactic expressions (which will tell us something new)
- Why proofs? Two kinds of inferences an agent might want to make:
  - Multiple percepts => conclusions about the world
  - Current state & operator => properties of next state

Propositional Logic Syntax

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for** (thing t = fizz; t == fuzz; t++){ … }

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for (thing t = fizz; t == fuzz; t++){ … }**
- *Colorless green ideas sleep furiously*.

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for (thing t = fizz; t == fuzz; t++){ … }**
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for (thing t = fizz; t == fuzz; t++){ ... }**
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

- <u>true</u> and <u>false</u> are sentences

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for** (thing t = fizz; t == fuzz; t++){ … }
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

- true and false are sentences
- Propositional variables are sentences: P,Q,R,Z

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for** (thing t = fizz; t == fuzz; t++){ … }
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

- <u>true</u> and <u>false</u> are sentences
- Propositional variables are sentences: P,Q,R,Z
- If ϕ and ψ are sentences, then so are
  (ϕ), $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$

Propositional Logic Syntax

Syntax: what you're allowed to write

- **for** (thing t = fizz; t == fuzz; t++){ ... }
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

- <u>true</u> and <u>false</u> are sentences
- Propositional variables are sentences: P,Q,R,Z
- If ϕ and ψ are sentences, then so are
  (ϕ), $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$
- Nothing else is a sentence

Precedence

$\neg$	highest
$\wedge$	
$\vee$	
$\rightarrow$	
$\leftrightarrow$	lowest

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

- Precedence rules enable "shorthand" form of sentences, but formally only the fully parenthesized form is legal.
- Syntactically ambiguous forms allowed in shorthand only when semantically equivalent: $A \wedge B \wedge C$ is equivalent to $(A \wedge B) \wedge C$ and $A \wedge (B \wedge C)$

Semantics

Semantics

- Meaning of a sentence is truth value {**t, f**}

Semantics

- Meaning of a sentence is truth value {**t, f**}
- Interpretation is an assignment of truth values to the propositional variables

  holds(ϕ,i) [Sentence ϕ is **t** in interpretation i]

Semantics

- Meaning of a sentence is truth value {**t, f**}
- Interpretation is an assignment of truth values to the propositional variables

  holds(ϕ, i) [Sentence ϕ is **t** in interpretation i]

  fails(ϕ, i) [Sentence ϕ is **f** in interpretation i]

Semantic Rules

Semantic Rules

- *holds*(true, i)   for all i

Semantic Rules

- *holds*(<u>true</u>, i)   for all i
- *fails*(<u>false</u>, i)   for all i

Semantic Rules

- *holds*(true, i) for all i
- *fails*(false, i) for all i
- *holds*($\neg\phi$, i) if and only if *fails*(ϕ, i)
  (negation)

Semantic Rules

- *holds*(<u>true</u>, i) for all i
- *fails*(<u>false</u>, i) for all i
- *holds*($\neg\phi$, i) if and only if *fails*(ϕ, i)
  (negation)
- *holds*($\phi \wedge \psi$, i) iff *holds*(ϕ, i) and *holds*(ψ,i)
  (conjunction)

Semantic Rules

- *holds*(<u>true</u>, i) for all i
- *fails*(<u>false</u>, i) for all i
- *holds*($\neg\phi$, i) if and only if *fails*(ϕ, i) (negation)
- *holds*($\phi \wedge \psi$, i) iff *holds*(ϕ, i) and *holds*(ψ,i) (conjunction)
- *holds*($\phi \vee \psi$, i) iff *holds*(ϕ, i) or *holds*(ψ,i) (disjunction)

Semantic Rules

- *holds*(<u>true</u>, i) for all i
- *fails*(<u>false</u>, i) for all i
- *holds*($\neg\phi$, i) if and only if *fails*(ϕ, i)
  (negation)
- *holds*($\phi \wedge \psi$, i) iff *holds*(ϕ, i) and *holds*(ψ,i)
  (conjunction)
- *holds*($\phi \vee \psi$, i) iff *holds*(ϕ, i) or *holds*(ψ,i)
  (disjunction)

- *holds*(P, i) iff i(P) = **t**
- *fails*(P, i) iff i(P) = **f**

Some important shorthand

Some important shorthand

- $\phi \rightarrow \psi \equiv \neg \phi \vee \psi$ (conditional, implication)

  antecedent $\rightarrow$ consequent

Some important shorthand

- $\phi \rightarrow \psi \equiv \neg \phi \vee \psi$ (conditional, implication)
  antecedent $\rightarrow$ consequent
- $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ (biconditional, equivalence)

Some important shorthand

- $\phi \rightarrow \psi \equiv \neg \phi \vee \psi$ (conditional, implication)
  antecedent $\rightarrow$ consequent
- $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ (biconditional, equivalence)

Truth Tables

P	Q	$\neg$ P	P $\wedge$ Q	P $\vee$ Q	P $\rightarrow$ Q	Q $\rightarrow$ P	P $\leftrightarrow$ Q
f	**f**	**t**	**f**	**f**	**t**	**t**	**t**
f	**t**	**t**	**f**	**t**	**t**	**f**	**f**
t	**f**	**f**	**f**	**t**	**f**	**t**	**f**
t	**t**	**f**	**t**	**t**	**t**	**t**	**t**

Some important shorthand

- $\phi \to \psi \equiv \neg \phi \vee \psi$ (conditional, implication)

  antecedent $\to$ consequent

- $\phi \leftrightarrow \psi \equiv (\phi \to \psi) \wedge (\psi \to \phi)$ (biconditional, equivalence)

Truth Tables

P	Q	$\neg$ P	P $\wedge$ Q	P $\vee$ Q	P $\to$ Q	Q $\to$ P	P $\leftrightarrow$ Q
f	**f**	**t**	**f**	**f**	**t**	**t**	**t**
f	**t**	**t**	**f**	**t**	**t**	**f**	**f**
t	**f**	**f**	**f**	**t**	**f**	**t**	**f**
t	**t**	**f**	**t**	**t**	**t**	**t**	**t**

Note that implication is not "causality", if P is **f** then P $\to$ Q is **t**

Terminology

Terminology

- A sentence is valid iff its truth value is **t** in all interpretations

  Valid sentences: true, $\neg$ false, $P \vee \neg P$

Terminology

- A sentence is valid iff its truth value is **t** in all interpretations

  Valid sentences: <u>true</u>, $\neg$ <u>false</u>, P $\vee$ $\neg$ P
- A sentence is satisfiable iff its truth value is **t** in at least one interpretation

  Satisfiable sentences: P, <u>true</u>, $\neg$ P

Terminology

- A sentence is valid iff its truth value is **t** in all interpretations

  Valid sentences: true, $\neg$ false, $P \vee \neg P$

- A sentence is satisfiable iff its truth value is **t** in at least one interpretation

  Satisfiable sentences: P, true, $\neg P$

- A sentence is unsatisfiable iff its truth value is **f** in all interpretations

  Unsatisfiable sentences: $P \wedge \neg P$, false, $\neg$ true

Terminology

- A sentence is valid iff its truth value is **t** in all interpretations

  Valid sentences: <u>true</u>, $\neg$ <u>false</u>, $P \vee \neg P$
- A sentence is satisfiable iff its truth value is **t** in at least one interpretation

  Satisfiable sentences: P, <u>true</u>, $\neg P$
- A sentence is unsatisfiable iff its truth value is **f** in all interpretations

  Unsatisfiable sentences: $P \wedge \neg P$, <u>false</u>, $\neg$ <u>true</u>

All are finitely decidable.

Examples

Examples

Sentence	Valid?	Interpretation that make sentence's truth value = **f**
smoke $\rightarrow$ smoke	valid	
smoke $\vee$ $\neg$smoke		

Examples

Sentence	Valid?	Interpretation that make sentence's truth value = **f**
smoke $\rightarrow$ smoke	valid	
smoke $\vee$ $\neg$smoke	valid	
smoke $\rightarrow$ fire	satisfiable, not valid	smoke = **t**, fire = **f**

Examples

Sentence	Valid?	Interpretation that make sentence's truth value = **f**
smoke $\rightarrow$ smoke smoke $\vee$ $\neg$smoke	valid	
smoke $\rightarrow$ fire	satisfiable, not valid	smoke = **t**, fire = **f**
$(s \rightarrow f) \rightarrow (\neg s \rightarrow \neg f)$	satisfiable, not valid	s = **f**, f = **t** $s \rightarrow f$ = **t**, $\neg s \rightarrow \neg f$ = **f**

Examples

Sentence	Valid?	Interpretation that make sentence's truth value = **f**
$smoke \rightarrow smoke$ $smoke \vee \neg smoke$	valid	
$smoke \rightarrow fire$	satisfiable, not valid	smoke = **t**, fire = **f**
$(s \rightarrow f) \rightarrow (\neg s \rightarrow \neg f)$	satisfiable, not valid	s = **f**, f = **t** $s \rightarrow f$ = **t**, $\neg s \rightarrow \neg f$ = **f**
contrapositive $(s \rightarrow f) \rightarrow (\neg f \rightarrow \neg s)$	valid	

Examples

Sentence	Valid?	Interpretation that make sentence's truth value = **f**
$smoke \rightarrow smoke$ $smoke \vee \neg smoke$	valid	
$smoke \rightarrow fire$	satisfiable, not valid	smoke = **t**, fire = **f**
$(s \rightarrow f) \rightarrow (\neg s \rightarrow \neg f)$	satisfiable, not valid	s = **f**, f = **t** $s \rightarrow f$ = **t**, $\neg s \rightarrow \neg f$ = **f**
contrapositive $(s \rightarrow f) \rightarrow (\neg f \rightarrow \neg s)$	valid	
$b \vee d \vee (b \rightarrow d)$ $b \vee d \vee \neg b \vee d$	valid	

Satisfiability

- Related to constraint satisfaction
- Given a sentence S, try to find an interpretation i such that *holds*(S,i)
- Analogous to finding an assignment of values to variables such that the constraints hold

Satisfiability

- Related to constraint satisfaction
- Given a sentence S, try to find an interpretation i such that *holds*(S,i)
- Analogous to finding an assignment of values to variables such that the constraints hold
- Brute force method: enumerate all interpretations and check

Satisfiability

- Related to constraint satisfaction
- Given a sentence S, try to find an interpretation i such that *holds*(S,i)
- Analogous to finding an assignment of values to variables such that the constraints hold
- Brute force method: enumerate all interpretations and check
- Better methods:
    - heuristic search
    - constraint propagation
    - stochastic search

Satisfiability problems

Satisfiability problems

- Scheduling nurses to work in a hospital
  - propositional variables represent, for example, that Pat is working on Tuesday at 2
  - constraints on the schedule are represented using logical expressions over the variables

Satisfiability problems

- Scheduling nurses to work in a hospital
  - propositional variables represent, for example, that Pat is working on Tuesday at 2
  - constraints on the schedule are represented using logical expressions over the variables
- Finding bugs in software
  - propositional variables represent state of the program
  - use logic to describe how the program works and to assert there is a bug
  - if the sentence is satisfiable, you've found a bug!

A Good Lecture?

A Good Lecture?

Imagine we knew that:

- If today is sunny, then Tomas will be happy ($S \rightarrow H$)
- If Tomas is happy, the lecture will be good ($H \rightarrow G$)
- Today is sunny (S)

Should we conclude that the lecture will be good?

Checking Interpretations

Checking Interpretations

S	H	G
t	t	t
t	t	f
t	f	t
t	f	f
f	t	t
f	t	f
f	f	t
f	f	f

Checking Interpretations

S	H	G	S→ H	H→G	S
t	t	t	t	t	t
t	t	f	t	f	t
t	f	t	f	t	t
t	f	f	f	t	t
f	t	t	t	t	f
f	t	f	t	f	f
f	f	t	t	t	f
f	f	f	t	t	f

Checking Interpretations

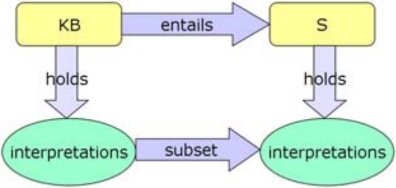
S	H	G	S →H	H→G	S	G
t	t	t	t	t	t	t
t	t	f	t	f	t	f
t	f	t	f	t	t	t
t	f	f	f	t	t	f
f	t	t	t	t	f	t
f	t	f	t	f	f	f
f	f	t	t	t	f	t
f	f	f	t	t	f	f

Adding a Variable

L	S	H	G	S→H	H→G	S	G
t	t	t	t	t	t	t	t
t	t	t	f	t	f	t	f
t	t	f	t	f	t	t	t
t	t	f	f	f	t	t	f
t	f	t	t	t	t	f	t
t	f	t	f	t	f	f	f
t	f	f	t	t	t	f	t
t	f	f	f	t	t	f	f
f	t	t	t	t	t	t	t
f	t	t	f	t	f	t	f
…	…	…					

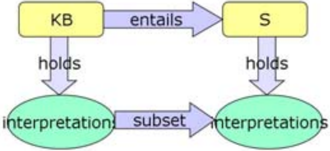
Entailment

A knowledge base (KB) *entails* a sentence S iff every interpretation that makes KB true also makes S true



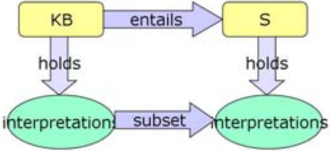
Computing Entailment

- enumerate all interpretations
- select those in which all elements of KB are true
- check to see if S is true in all of those interpretations



Computing Entailment

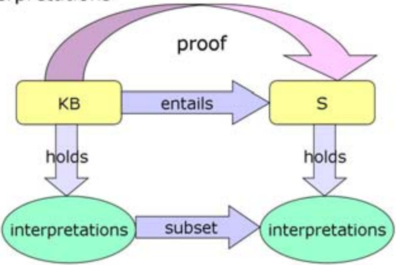
- enumerate all interpretations
- select those in which all elements of KB are true
- check to see if S is true in all of those interpretations



Way too many interpretations, in general!!

Entailment and Proof

A proof is a way to test whether a KB entails a sentence, without enumerating all possible interpretations



Proof

Proof

- Proof is a sequence of sentences

Proof

- Proof is a sequence of sentences
- First ones are premises (KB)

Proof

- Proof is a sequence of sentences
- First ones are premises (KB)
- Then, you can write down on the next line the result of applying an inference rule to previous lines

Proof

- Proof is a sequence of sentences
- First ones are premises (KB)
- Then, you can write down on the next line the result of applying an inference rule to previous lines
- When S is on a line, you have proved S from KB

Proof

- Proof is a sequence of sentences
- First ones are premises (KB)
- Then, you can write down on the next line the result of applying an inference rule to previous lines
- When S is on a line, you have proved S from KB

- If inference rules are *sound*, then any S you can prove from KB is entailed by KB

Proof

- Proof is a sequence of sentences
- First ones are premises (KB)
- Then, you can write down on the next line the result of applying an inference rule to previous lines
- When S is on a line, you have proved S from KB

- If inference rules are *sound*, then any S you can prove from KB is entailed by KB

- If inference rules are *complete*, then any S that is entailed by KB can be proved from KB

Natural Deduction

Some inference rules:

Natural Deduction

Some inference rules:

$$\frac{\begin{array}{c}\alpha \rightarrow \beta \\ \alpha\end{array}}{\beta}$$

Modus ponens

Natural Deduction

Some inference rules:

$$\frac{\begin{array}{c}\alpha \rightarrow \beta \\ \alpha\end{array}}{\beta}$$

Modus ponens

$$\frac{\begin{array}{c}\alpha \rightarrow \beta \\ \neg \beta\end{array}}{\neg \alpha}$$

Modus tolens

Natural Deduction

Some inference rules:

Modus ponens	Modus tolens	And-introduction
$\dfrac{\alpha \to \beta \quad \alpha}{\beta}$	$\dfrac{\alpha \to \beta \quad \neg \beta}{\neg \alpha}$	$\dfrac{\alpha \quad \beta}{\alpha \wedge \beta}$

Natural Deduction

Some inference rules:

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

Modus ponens

$$\frac{\alpha \rightarrow \beta \quad \neg \beta}{\neg \alpha}$$

Modus tolens

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

And-introduction

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

And-elimination

Natural deduction example

Prove S

Step	Formula	Derivation

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim
5	R	4,2 Modus Ponens

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim
5	R	4,2 Modus Ponens
6	Q	1 And-Elim

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim
5	R	4,2 Modus Ponens
6	Q	1 And-Elim
7	$Q \wedge R$	5,6 And-Intro

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim
5	R	4,2 Modus Ponens
6	Q	1 And-Elim
7	$Q \wedge R$	5,6 And-Intro
8	S	7,3 Modus Ponens

Proof systems

- There are many natural deduction systems; they are typically "proof checkers", sound but not complete

Proof systems

- There are many natural deduction systems; they are typically "proof checkers", sound but not complete
- Natural deduction uses lots of inference rules which introduces a large branching factor in the search for a proof.

Proof systems

- There are many natural deduction systems; they are typically "proof checkers", sound but not complete
- Natural deduction uses lots of inference rules which introduces a large branching factor in the search for a proof.
- In general, you need to do "proof by cases" which introduces even more branching.

Prove R

1	$P \vee Q$
2	$Q \rightarrow R$
3	$P \rightarrow R$

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Single inference rule is a sound and complete proof system

- Requires all sentences to be converted to conjunctive normal form

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

  - $(A \vee B \vee \neg C)$ is a clause

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

  - $(A \vee B \vee \neg C)$ is a clause, which is a disjunction of literals
  - A, B, and ¬ C are literals

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

  - $(A \vee B \vee \neg C)$ is a clause, which is a disjunction of literals
  - A, B, and ¬ C are literals, each of which is a variable or the negation of a variable.

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

  - $(A \vee B \vee \neg C)$ is a clause, which is a disjunction of literals
  - A, B, and ¬ C are literals, each of which is a variable or the negation of a variable.
  - Each clause is a requirement that must be satisfied and can be satisfied in multiple ways

Conjunctive Normal Form

- Conjunctive normal form (CNF) formulas:

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

  - $(A \vee B \vee \neg C)$ is a clause, which is a disjunction of literals
  - A, B, and ¬ C are literals, each of which is a variable or the negation of a variable.
  - Each clause is a requirement that must be satisfied and can be satisfied in multiple ways
  - Every sentence in propositional logic can be written in CNF

Converting to CNF

Converting to CNF

1. Eliminate arrows using definitions

Converting to CNF

1. Eliminate arrows using definitions
2. Drive in negations using De Morgan's Laws

$$\neg(\phi \vee \varphi) \equiv \neg\phi \wedge \neg\varphi$$
$$\neg(\phi \wedge \varphi) \equiv \neg\phi \vee \neg\varphi$$

Converting to CNF

1. Eliminate arrows using definitions
2. Drive in negations using De Morgan's Laws

$$\neg(\phi \vee \varphi) \equiv \neg\phi \wedge \neg\varphi$$
$$\neg(\phi \wedge \varphi) \equiv \neg\phi \vee \neg\varphi$$

3. Distribute **or** over **and**

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

Converting to CNF

1. Eliminate arrows using definitions
2. Drive in negations using De Morgan's Laws

$$\neg(\phi \vee \varphi) \equiv \neg\phi \wedge \neg\varphi$$
$$\neg(\phi \wedge \varphi) \equiv \neg\phi \vee \neg\varphi$$

3. Distribute **or** over **and**

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

4. Every sentence can be converted to CNF, but it may grow exponentially in size

CNF Conversion Example

$$(A \vee B) \rightarrow (C \rightarrow D)$$

CNF Conversion Example

$$(A \vee B) \rightarrow (C \rightarrow D)$$

1. Eliminate arrows

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

CNF Conversion Example

$$(A \vee B) \rightarrow (C \rightarrow D)$$

1. Eliminate arrows

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

2. Drive in negations

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

CNF Conversion Example

$$(A \vee B) \rightarrow (C \rightarrow D)$$

1. Eliminate arrows

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

2. Drive in negations

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

3. Distribute

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

Propositional Resolution

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Resolution refutation:

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF
  - Negate the desired conclusion (converted to CNF)

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF
  - Negate the desired conclusion (converted to CNF)
  - Apply resolution rule until either
    - Derive false (a contradiction)
    - Can't apply any more

Propositional Resolution

- Resolution rule:

$$\frac{\begin{array}{l}\alpha \vee \beta \\ \neg\beta \vee \gamma\end{array}}{\alpha \vee \gamma}$$

- Resolution refutation:
  - Convert all sentences to CNF
  - Negate the desired conclusion (converted to CNF)
  - Apply resolution rule until either
    - Derive false (a contradiction)
    - Can't apply any more
- Resolution refutation is sound and complete
  - If we derive a contradiction, then the conclusion follows from the axioms
  - If we can't apply any more, then the conclusion cannot be proved from the axioms.

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given

Propositional Resolution Example

Prove R

1	P v Q
2	$P \to R$
3	$Q \to R$

Step	Formula	Derivation
1	P v Q	Given
2	¬ P v R	Given
3	¬ Q v R	Given

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \to R$
3	$Q \to R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion

Propositional Resolution Example

Prove R

1	P v Q
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	P v Q	Given
2	$\neg$ P v R	Given
3	$\neg$ Q v R	Given
4	$\neg$ R	Negated conclusion
5	Q v R	1,2

Propositional Resolution Example

Prove R

1	P v Q
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	P v Q	Given
2	¬ P v R	Given
3	¬ Q v R	Given
4	¬ R	Negated conclusion
5	Q v R	1,2
6	¬ P	2,4

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion
5	$Q \vee R$	1,2
6	$\neg P$	2,4
7	$\neg Q$	3,4

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion
5	$Q \vee R$	1,2
6	$\neg P$	2,4
7	$\neg Q$	3,4
8	R	5,7

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion
5	$Q \vee R$	1,2
6	$\neg P$	2,4
7	$\neg Q$	3,4
8	R	5,7
9	•	4,8

Propositional Resolution Example

Prove R

1	P v Q
2	P → R
3	Q → R

Step	Formula	Derivation
1	P v Q	Given
2	¬ P v R	Given
3	¬ Q v R	Given
4	¬ R	Negated conclusion
5	Q v R	1,2
6	¬ P	2,4
7	¬ Q	3,4
8	R	5,7
9	•	4,8

false v R

¬ R v false

false v false

Propositional Resolution Example

Prove R

1	$P \vee Q$
2	$P \rightarrow R$
3	$Q \rightarrow R$

Step	Formula	Derivation
1	$P \vee Q$	Given
2	$\neg P \vee R$	Given
3	$\neg Q \vee R$	Given
4	$\neg R$	Negated conclusion
5	$Q \vee R$	1,2
6	$\neg P$	2,4
7	$\neg Q$	3,4
8	R	5,7
9	•	4,8

$$\frac{\begin{array}{l}\text{false} \vee R \\ \neg R \vee \text{false}\end{array}}{\text{false} \vee \text{false}}$$

The Power of False

Prove Z

1	P
2	$\neg$ P

Step	Formula	Derivation

The Power of False

Prove Z

1	P
2	¬ P

Step	Formula	Derivation
1	P	Given
2	¬ P	Given
3	¬ Z	Negated conclusion

The Power of False

Prove Z

1	P
2	¬ P

Step	Formula	Derivation
1	P	Given
2	¬ P	Given
3	¬ Z	Negated conclusion
4	•	1,2

The Power of False

Prove Z

1	P
2	$\neg$ P

Step	Formula	Derivation
1	P	Given
2	$\neg$ P	Given
3	$\neg$ Z	Negated conclusion
4	•	1,2

Note that $(P \wedge \neg P) \rightarrow Z$ is valid

The Power of False

Prove Z

1	P
2	¬ P

Step	Formula	Derivation
1	P	Given
2	¬ P	Given
3	¬ Z	Negated conclusion
4	•	1,2

Note that $(P \wedge \neg P) \rightarrow Z$ is valid

Any conclusion follows from a contradiction – and so strict logic systems are very brittle.

Example Problem

Convert to CNF

Prove R

1	$(P \rightarrow Q) \rightarrow Q$
2	$(P \rightarrow P) \rightarrow R$
3	$(R \rightarrow S) \rightarrow \neg(S \rightarrow Q)$

Example Problem

Prove R

1	$(P \to Q) \to Q$
2	$(P \to P) \to R$
3	$(R \to S) \to \neg(S \to Q)$

Convert to CNF

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

Example Problem

Prove R

1	$(P \rightarrow Q) \rightarrow Q$
2	$(P \rightarrow P) \rightarrow R$
3	$(R \rightarrow S) \rightarrow \neg(S \rightarrow Q)$

Convert to CNF

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

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

Example Problem

Prove R

1	$(P \to Q) \to Q$
2	$(P \to P) \to R$
3	$(R \to S) \to \neg(S \to Q)$

Convert to CNF

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

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

- $\neg(\neg R \vee S) \vee \neg(\neg S \vee Q)$
- $(R \wedge \neg S) \vee (S \wedge \neg Q)$
- $(R \vee S) \wedge (\neg S \vee S) \wedge (R \vee \neg Q) \wedge (\neg S \vee \neg Q)$
- $(R \vee S) \wedge (R \vee \neg Q) \wedge (\neg S \vee \neg Q)$

Resolution Proof Example

Prove R

1	$(P \to Q) \to Q$
2	$(P \to P) \to R$
3	$(R \to S)$ $\to \neg (S \to Q)$

1	$P \vee Q$	
2	$P \vee R$	
3	$\neg P \vee R$	
4	$R \vee S$	
5	$R \vee \neg Q$	
6	$\neg S \vee \neg Q$	
7	$\neg R$	Neg

Resolution Proof Example

Prove R

1	$(P \to Q) \to Q$
2	$(P \to P) \to R$
3	$(R \to S) \to \neg (S \to Q)$

1	$P \vee Q$	
2	$P \vee R$	
3	$\neg P \vee R$	
4	$R \vee S$	
5	$R \vee \neg Q$	
6	$\neg S \vee \neg Q$	
7	$\neg R$	Neg
8	S	4,7
9	$\neg Q$	6,8
10	P	1,9
11	R	3,10
12	•	7,11

Proof Strategies

Proof Strategies

- Unit preference: prefer a resolution step involving an unit clause (clause with one literal).
  - Produces a shorter clause – which is good since we are trying to produce a zero-length clause, that is, a contradiction.
- Set of support: Choose a resolution involving the negated goal or any clause derived from the negated goal.
  - We're trying to produce a contradiction that follows from the negated goal, so these are "relevant" clauses.
  - If a contradiction exists, one can find one using the set-of-support strategy.