Lecture - 5

Propositional Logic: Reasoning

Previously

Name	Symbol	Notation	Definition
Conjunction	^	P^Q	AND
Disjunction	⌄	P⌄Q	OR
Negation	~	~P	NOT
Implication/Conditional	->	P -> Q	IF / THEN
Biconditional	<->	P <-> Q	IF AND ONLY IF

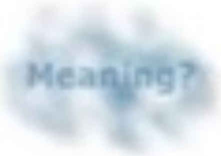
p	q	~p	~p ∨ q
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

$$\underbrace{\underbrace{a \Rightarrow b}_{\text{abstr. prop.}} \stackrel{val}{==} \underbrace{\neg a \vee b}_{\text{abstr. prop.}}}_{\text{meta-formula}}$$

$$\begin{aligned} & \neg P \Rightarrow Q \\ \stackrel{val}{==} \quad & \{ \text{ Implication } \} \\ & \neg\neg P \vee Q \\ \stackrel{val}{==} \quad & \{ \text{ Double Negation } \} \\ & P \vee Q \end{aligned}$$

- What is Propositional Logic?
- Propositional Language
- Propositional Truth
- Equivalence
- Logical Calculation

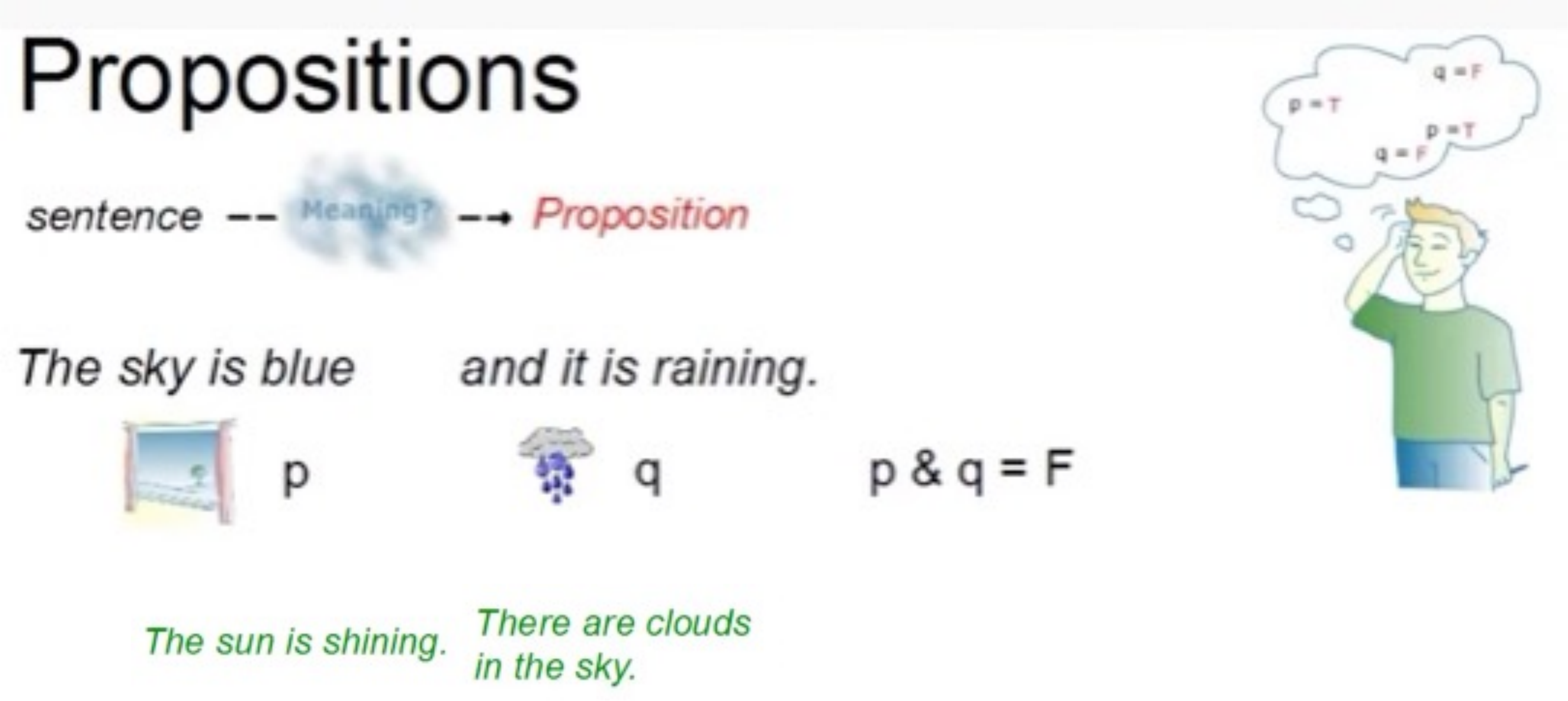
Propositions

sentence -- 

--→ *Proposition*

The sky is blue *and it is raining.*

 p  q p & q = F



Logical inference - from the single proposition which is true.

Lecture Outline

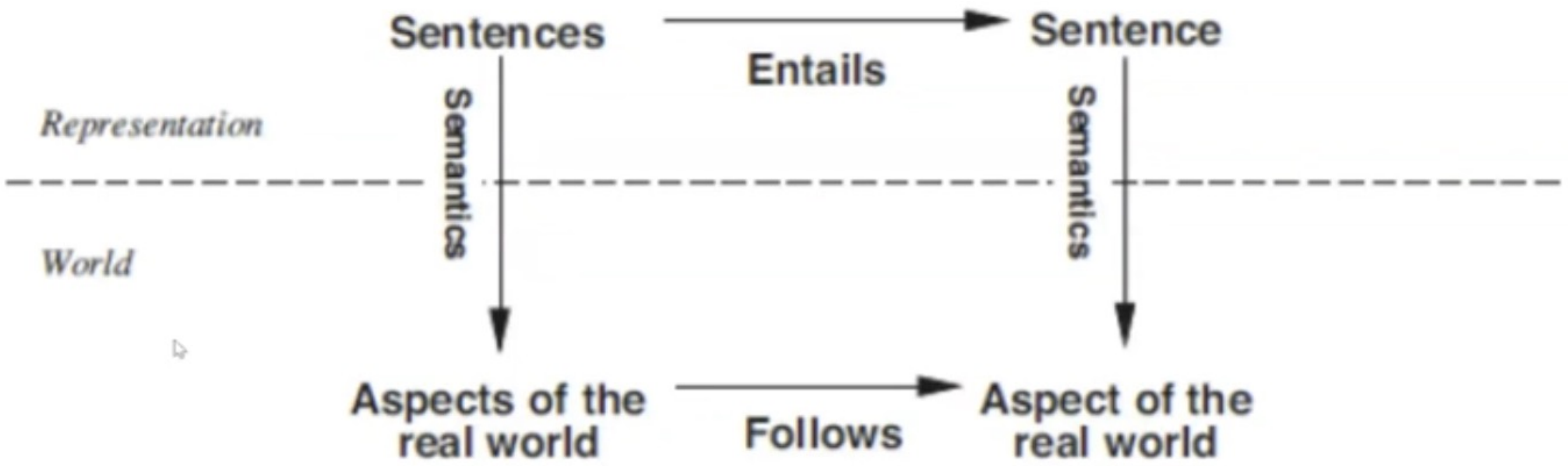
- Entailment
- Arguments
- Inference Rules
- Deductive Reasoning from a Knowledge Base

Why Reasoning?

- With an understanding of the syntax and semantics of propositional logic for knowledge representation…
  - We explore basic reasoning with logical inference
- Leverage a ‘represented’ knowledge base to draw conclusions
- Reasoning and inference will be recurring themes throughout this course

Representation to Real World

- Logic operates on propositions (i.e. statements) that are representations of real world objects/states
- In AI we are interested in taking existing knowledge and deriving new knowledge or answering questions (ultimately in the real world)
- In propositional logic, this means showing that from a knowledge base, (i.e. a possibly extensive propositional logic formula), a formula (conclusion) follows.



Logic: Model

- Not the same as a ‘model’ from machine learning!

- A model is an interpretation of a proposition that is true

- Notation:
  - When proposition (f) is true in model (M) we say: M satisfies f
  - M(f): means, the set of all models where f is true

- Example: For $f = A \wedge B$

  $M(f) = \{A = 1, B = 1\}$.

- For $f = A \leftrightarrow B$

  $M(f) = \{A=1, B=1\}; \{A=0; B=0\}$

Logic: Model

- Not the same as a 'model' from machine learning!

- A model is an interpretation of a proposition that is true

- Notation:
  - When proposition (f) is true in model (M) we say: M satisfies f
  - M(f): means, the set of all models where f is true

- Example: For f= $A \wedge B$

  $M(f) = \{ A = 1, B = 1\}$.

- For f = $A \leftrightarrow B$

  $M(f) = \{A=1, B=1\}; \{A = 0; B = 0\}$

Tautology: Formula f such that M(f) = All possible models. ("True in all possible worlds")

Example: $A \vee \neg A$.
(True whether A =0 or A=1!)

Contradiction: Formula f such that M(f) = Empty set. ("False in all possible worlds.")

Example: $A \wedge \neg A$.
(False whether A =0 or A=1!)

Knowledge Base

- Knowledge base (KB) : Set of propositions $\{f_1, f_2, \ldots, f_n\}$
- M(KB) = All possible models for $f_1 \wedge f_2 \wedge \ldots \wedge f_n$

Example:

- Variables: R, S, C ("Rainy", "Sunny," "Cloudy")
- KB:

$R \vee S \vee C$	("It is either Rainy or Sunny or Cloudy.")
$R \rightarrow C \wedge \neg S$	("If it is Rainy then it is Cloudy and not Sunny.")
$C \leftrightarrow \neg S$	("If it is Cloudy then it is not Sunny, and vice versa.")

- Example models for KB: {R=1, S =0, C =1}; {R =0, C=1, S=0}; {R=0, C=0, S=1}

Entailment

In pragmatics (linguistics), entailment is the relationship between two sentences where the truth of one (A) requires the truth of the other (B). For example, the sentence (A) The president was assassinated. entails (B) The president is dead.

Entailment is a concept that refers to a specific kind of relationship between two sentences. More specifically, entailment means that if one sentence is true, then another sentence would also have to be true: the second sentence would be entailed by the first sentence.

Entailment

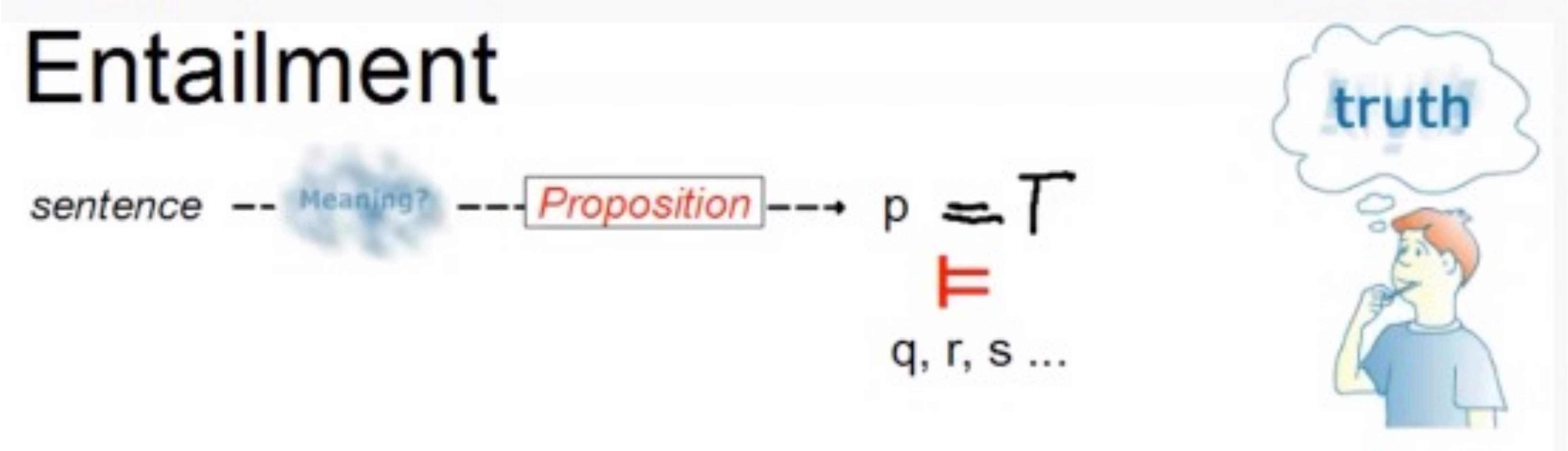
- Aka: Logical consequence $\boxed{\models}$
- One statement logically *follows* from one or more propositions
  - Knowledge base = set of propositions or a single proposition that asserts all individual sentences
  - Entailments are the basis for making maximally reliable inferences from a knowledge base

- Examples:

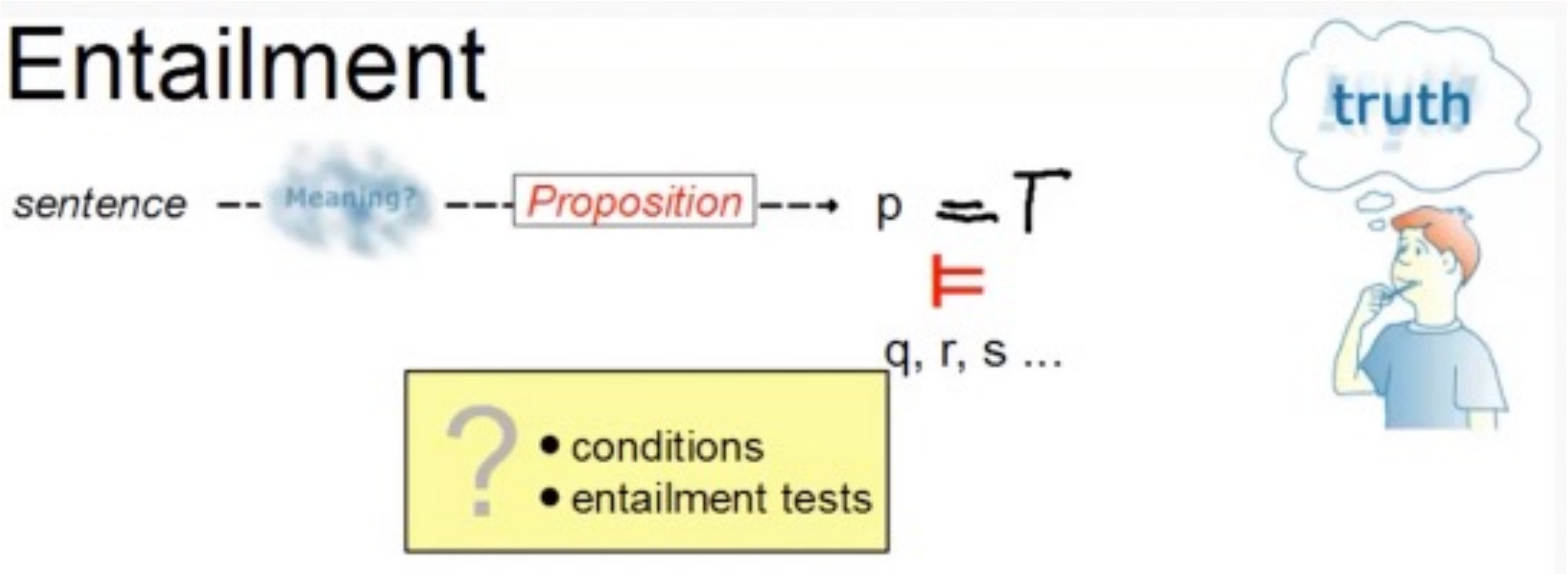
$$\{p\} \models p \vee q$$

$$\{p\} \not\models p \wedge q \quad \longleftarrow \text{Not entailed}$$

$$\{p, q\} \models p \wedge q$$



Entailment is defined as any true inference from a true proposition



For a proposition to be entail another proposition

- A set of conditions to be satisficed
- Entailments can be tested

Entailment $\models$

Brutus killed Caesar.
p = Brutus killed Caesar
q = Caesar died.

$p \models q$

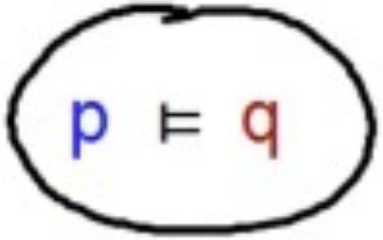
p	q	$p \models q$
T	T	T
F	T v F	F

Entailment ⊨

Brutus killed Caesar.

p = Brutus killed Caesar

q = Caesar died.



p	q	p ⊨ q
T	T	T
F	T v F	F

q	p	q ⊨ p
T	T v F	T
F	F	F

Entailment is a logical relationship between two proposition such that truth of first proposition **p** guarantees the truth of 2nd proposition **q.**

Falsity of **q** guarantee the falsity of **p**.

Entailment $\vDash$

All dogs are purple.

p = All dogs are purple.

q = My dog is purple.

$p \vDash q$

$p \vDash q$, iff $p = T$ & $q = T$

Entailment Test

Step ① p

Step ② $p \models q$

Step ③ $\neg q$

Step ④ $p \ \& \neg q$

→ contradiction $p \models q$

→ no contradiction $p \not\models q$

Entailment Test - Example 1

1. p All dogs are purple.
2. $p \vdash$ $\boxed{q \text{ My dog is purple.}}$
3. $\neg q$ My dog is NOT purple.
4. $p \;\&\; \neg q$

All dogs are purple and/but my dog is NOT purple.

contradiction: $p \vdash q$

Entailment Test - Example 2

1. p All dogs are purple.
2. $p \vDash$ $\boxed{q \text{ My dog likes cats.}}$
3. $\neg q$ My dog does NOT like cats.
4. $p \ \& \neg q$

All dogs are purple and/but my dog does NOT like cats.

no contradiction: $p \nvDash q$

For now, for simplicity, we will focus on entailments dealing with a single proposition!
[From which a conclusion follows]

We will return to multiple propositions from a knowledge base, when move on to arguments.

Equivalence vs Entailment

Recall:

$$P \overset{val}{==} Q \quad \text{means} \quad \begin{cases} \text{(a) whenever } P \text{ is 1, then also } Q \text{ is 1} \\ \text{(b) whenever } Q \text{ is 1, then also } P \text{ is 1} \end{cases}$$

Define:

$$\boxed{P \overset{val}{\models} Q \quad \text{means} \quad \{ \text{ (a) whenever } P \text{ is 1, then also } Q \text{ is 1}}$$

Pronounce $P \overset{val}{\models} Q$ as "*P is stronger than Q*."

Entailment is actually a Oneway equivalance

Equivalence vs Entailment

Recall:

$$P \overset{val}{=\!=} Q \quad \text{means} \quad \begin{cases} \text{(a) whenever } P \text{ is } 1 \text{, then also } Q \text{ is } 1 \\ \text{(b) whenever } Q \text{ is } 1 \text{, then also } P \text{ is } 1 \end{cases}$$

Define:

$$\boxed{P \overset{val}{\models} Q \quad \text{means} \quad \{ \text{ (a) whenever } P \text{ is } 1 \text{, then also } Q \text{ is } 1}$$

Pronounce $P \overset{val}{\models} Q$ as "*P is stronger than Q*."

- P (our KB) entails a formula Q (i.e. Q follows from KB), if every model of P is also a model of Q

$$P \models Q$$

Equivalence vs Entailment

Recall:

$$P \overset{val}{==} Q \quad \text{means} \quad \begin{cases} \text{(a) whenever } P \text{ is } 1, \text{ then also } Q \text{ is } 1 \\ \text{(b) whenever } Q \text{ is } 1, \text{ then also } P \text{ is } 1 \end{cases}$$

Define:

$$\boxed{P \overset{val}{\models} Q \quad \text{means} \quad \{\ \text{(a) whenever } P \text{ is } 1, \text{ then also } Q \text{ is } 1}$$

Pronounce $P \overset{val}{\models} Q$ as "*P is stronger than Q.*"

- P (our KB) entails a formula Q (i.e. Q follows from KB), if every model of P is also a model of Q

$$P \models Q$$

P	Q
$1 \longrightarrow$	1
0	$1/0$

$P \overset{val}{\models} Q$: 1s are *carried over* from P to Q.

Determining Entailment from Truth Table

- **Example 1:**

$$\neg P \left\{ \begin{array}{c} \overset{val}{\models} ? \\ \overset{val}{=\!\!|} ? \end{array} \right\} P \Rightarrow Q$$

P	Q	$\neg P$	$P \Rightarrow Q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

extra true

So $\neg P$ is stronger than $P \Rightarrow Q$ (i.e., $\neg P \overset{val}{\models} P \Rightarrow Q$).

A 'stronger' proposition entails a weaker one (i.e. infers it)

Determining Entailment from Truth Table

- **Example 2:**

$$P \Rightarrow Q \left\{ \begin{array}{c} \overset{val}{\models} \; ? \\ \overset{val}{\Dashv} \; ? \end{array} \right\} P \vee Q$$

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

So $P \Rightarrow Q$ and $P \vee Q$ are incomparable.

Standard Weakenings of Entailment

- Entailment transformations

∧-∨-weakening:

$$P \wedge Q \overset{val}{\models} P$$

$$P \overset{val}{\models} P \vee Q$$

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

Also:

$$P \wedge Q \overset{val}{\models} Q \quad \text{and}$$

$$Q \overset{val}{\models} P \vee Q \ .$$

Standard Weakenings of Entailment

- Entailment transformations

∧-∨-weakening:

$$P \wedge Q \stackrel{val}{\models} P$$
$$P \stackrel{val}{\models} P \vee Q$$

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

Also:

$$P \wedge Q \stackrel{val}{\models} Q \quad \text{and}$$
$$Q \stackrel{val}{\models} P \vee Q \ .$$

Extremes:

$$\texttt{False} \stackrel{val}{\models} P$$
$$P \stackrel{val}{\models} \texttt{True}$$

`False` is strongest of all
`True` is weakest of all

Basic Properties of Entailment

$P \overset{val}{\models} P$.

If $P \overset{val}{\models} Q$, then $Q \overset{val}{=\!\!|} P$, and vice versa.

If $P \overset{val}{\models} Q$ and $Q \overset{val}{\models} R$, then $P \overset{val}{\models} R$.

Basic Properties of Entailment

$P \stackrel{val}{\models} P$.

If $P \stackrel{val}{\models} Q$, then $Q \stackrel{val}{=\mid} P$, and vice versa.

If $P \stackrel{val}{\models} Q$ and $Q \stackrel{val}{\models} R$, then $P \stackrel{val}{\models} R$.

$P \stackrel{val}{=\!=} Q$ if, and only if, $P \stackrel{val}{\models} Q$ and $P \stackrel{val}{=\mid} Q$.

So, if you need to prove $P \stackrel{val}{\models} Q$ or $P \stackrel{val}{=\mid} Q$ by a calculation, then it is enough to prove $P \stackrel{val}{=\!=} Q$.

But $P \stackrel{val}{\models} Q$ (or $P \stackrel{val}{=\mid} Q$) alone is not enough to conclude $P \stackrel{val}{=\!=} Q$!

Basic Properties of Entailment

$P \overset{val}{\models} P.$

If $P \overset{val}{\models} Q$, then $Q \overset{val}{=\!\!\mid} P$, and vice versa.

If $P \overset{val}{\models} Q$ and $Q \overset{val}{\models} R$, then $P \overset{val}{\models} R$.

$P \overset{val}{=\!=} Q$ if, and only if, $P \overset{val}{\models} Q$ and $P \overset{val}{=\!\!\mid} Q$.

So, if you need to prove $P \overset{val}{\models} Q$ or $P \overset{val}{=\!\!\mid} Q$ by a calculation, then it is enough to prove $P \overset{val}{=\!=} Q$.

But $P \overset{val}{\models} Q$ (or $P \overset{val}{=\!\!\mid} Q$) alone is not enough to conclude $P \overset{val}{=\!=} Q$!

$P \overset{val}{\models} Q$ if, and only if, $P \Rightarrow Q$ is a tautology

Deduction theorem!

Substitution Rule for Entailment

The Substitution Rule also works for $\overset{val}{\models}$ and $\overset{val}{=\!\!|}$:

$$\boxed{\text{SUBSTITUTION PRESERVES WEAKENING/STRENGHTENING}}$$

Example

We have the following valid weakening:

$$P \wedge Q \overset{val}{\models} P \vee R$$

and hence, according to the Substitution Rule, if we substitute $(Q \Rightarrow R)$ for P and $(P \vee Q)$ for Q, we get another valid weakening:

$$(Q \Rightarrow R) \wedge (P \vee Q) \overset{val}{\models} (Q \Rightarrow R) \vee R \ .$$

Leibniz's Rule for Entailment?

Recall Leibniz's Rule for making new equivalences:

From the valid equivalence

$$P \Rightarrow Q \overset{val}{=\!=} \neg P \vee Q$$

we can *make new valid equivalences* by replacing $P \Rightarrow Q$ in some complex formula by $\neg P \vee Q$, for instance:

$$(\neg P \wedge (P \Rightarrow Q)) \vee R \overset{val}{=\!=} (\neg P \wedge (\neg P \vee Q)) \vee R$$

Leibniz's Rule for Entailment? No!

Recall Leibniz's Rule for making new equivalences:

From the valid equivalence

$$P \Rightarrow Q \stackrel{val}{==} \neg P \vee Q$$

we can *make new valid equivalences* by replacing $P \Rightarrow Q$ in some complex formula by $\neg P \vee Q$, for instance:

$$(\neg P \wedge (P \Rightarrow Q)) \vee R \stackrel{val}{==} (\neg P \wedge (\neg P \vee Q)) \vee R$$

Examples

Note that, by $\wedge$-$\vee$-weakening, $P \wedge Q \stackrel{val}{\models} P \vee Q$. Now consider:

1. $\neg(P \wedge Q) \stackrel{val}{\not\models} \neg(P \vee Q)$;
2. $R \Rightarrow (P \wedge Q) \stackrel{val}{\models} R \Rightarrow (P \vee Q)$; and
3. $(P \wedge Q) \Rightarrow R \stackrel{val}{\not\models} (P \vee Q) \Rightarrow R$.

Conclusion: replacing $\stackrel{val}{==}$ by $\stackrel{val}{\models}$ does not yield a valid rule!

Monotonicity

We do have the following weaker variant of Leibniz's Rule:

Monotonicity:

(1) If $P \stackrel{val}{\models} Q$, then $P \wedge R \stackrel{val}{\models} Q \wedge R$

(2) If $P \stackrel{val}{\models} Q$, then $P \vee R \stackrel{val}{\models} Q \vee R$

- The syllogism:
    - "All men are mortal. Socrates is a man. Therefore Socrates is mortal."
- Can be weakened by adding a premise:
    - "All men are mortal. Socrates is a man. Cows produce milk. Therefore Socrates is mortal."

New premise

- The validity of the original conclusion is not changed by the addition of premises

Example of Using Entailment in a Proof

Prove with a calculation that $\neg(P \Rightarrow Q) \Rightarrow (\neg Q \wedge (P \vee R))$ is a tautology.

Monotonicity:

(1) If $P \overset{val}{\models} Q$, then $P \wedge R \overset{val}{\models} Q \wedge R$

(2) If $P \overset{val}{\models} Q$, then $P \vee R \overset{val}{\models} Q \vee R$

Summary: Equivalences for Connectives

Distributivity:

$P \wedge (Q \vee R) \stackrel{val}{=\!=} (P \wedge Q) \vee (P \wedge R)$,

$P \vee (Q \wedge R) \stackrel{val}{=\!=} (P \vee Q) \wedge (P \vee R)$

Inversion:

$\neg\texttt{True} \stackrel{val}{=\!=} \texttt{False}$,

$\neg\texttt{False} \stackrel{val}{=\!=} \texttt{True}$

Contradiction:

$P \wedge \neg P \stackrel{val}{=\!=} \texttt{False}$

Excluded Middle:

$P \vee \neg P \stackrel{val}{=\!=} \texttt{True}$

Bi-implication:

$P \Leftrightarrow Q \stackrel{val}{=\!=} (P \Rightarrow Q) \wedge (Q \Rightarrow P)$

Double Negation:

$\neg\neg P \stackrel{val}{=\!=} P$

Negation:

$\neg P \stackrel{val}{=\!=} P \Rightarrow \texttt{False}$

Associativity:

$(P \wedge Q) \wedge R \stackrel{val}{=\!=} P \wedge (Q \wedge R)$,

$(P \vee Q) \vee R \stackrel{val}{=\!=} P \vee (Q \vee R)$,

$(P \Leftrightarrow Q) \Leftrightarrow R \stackrel{val}{=\!=} P \Leftrightarrow (Q \Leftrightarrow R)$

De Morgan:

$\neg(P \wedge Q) \stackrel{val}{=\!=} \neg P \vee \neg Q$,

$\neg(P \vee Q) \stackrel{val}{=\!=} \neg P \wedge \neg Q$

Implication:

$P \Rightarrow Q \stackrel{val}{=\!=} \neg P \vee Q$

Contraposition:

$P \Rightarrow Q \stackrel{val}{=\!=} \neg Q \Rightarrow \neg P$

Self-equivalence:

$P \Leftrightarrow P \stackrel{val}{=\!=} \texttt{True}$

True/False-elimination:

$P \wedge \texttt{True} \stackrel{val}{=\!=} P$,

$P \wedge \texttt{False} \stackrel{val}{=\!=} \texttt{False}$,

$P \vee \texttt{True} \stackrel{val}{=\!=} \texttt{True}$,

$P \vee \texttt{False} \stackrel{val}{=\!=} P$

Commutativity:

$P \wedge Q \stackrel{val}{=\!=} Q \wedge P$,

$P \vee Q \stackrel{val}{=\!=} Q \vee P$,

$P \Leftrightarrow Q \stackrel{val}{=\!=} Q \Leftrightarrow P$

Idempotence:

$P \wedge P \stackrel{val}{=\!=} P$,

$P \vee P \stackrel{val}{=\!=} P$

Absorption Laws

$p \vee (p \wedge q) \equiv p$

$p \wedge (p \vee q) \equiv p$

Example of Using Entailment in a Proof

Prove with a calculation that $\neg(P \Rightarrow Q) \Rightarrow (\neg Q \wedge (P \vee R))$ is a tautology.

We know... $P \overset{val}{\models} Q$ if, and only if, $P \Rightarrow Q$ is a tautology.

Monotonicity:

(1) If $P \overset{val}{\models} Q$, then $P \wedge R \overset{val}{\models} Q \wedge R$

(2) If $P \overset{val}{\models} Q$, then $P \vee R \overset{val}{\models} Q \vee R$

Example of Using Entailment in a Proof

Prove with a calculation that $\neg(P \Rightarrow Q) \Rightarrow (\neg Q \wedge (P \vee R))$ is a tautology.

We know... $\boxed{P \overset{val}{\models} Q \text{ if, and only if, } P \Rightarrow Q \text{ is a tautology.}}$

First, we establish that $\neg(P \Rightarrow Q) \overset{val}{\models} \neg Q \wedge (P \vee R)$:

$$
\begin{array}{ll}
 & \neg(P \Rightarrow Q) \\
\overset{val}{=\!=} & \{ \text{ Implication } \} \\
 & \neg(\neg P \vee Q) \\
\overset{val}{=\!=} & \{ \text{ De Morgan } \} \\
 & \neg\neg P \wedge \neg Q \\
\overset{val}{=\!=} & \{ \text{ Double Negation } \} \\
 & P \wedge \neg Q \\
\overset{val}{\models} & \{ \wedge\text{-}\vee\text{-weakening + Monotonicity } \} \\
 & \neg Q \wedge (P \vee R)
\end{array}
$$

Monotonicity:

(1) If $P \overset{val}{\models} Q$, then $P \wedge R \overset{val}{\models} Q \wedge R$

(2) If $P \overset{val}{\models} Q$, then $P \vee R \overset{val}{\models} Q \vee R$

Example of Using Entailment in a Proof

Prove with a calculation that $\neg(P \Rightarrow Q) \Rightarrow (\neg Q \wedge (P \vee R))$ is a tautology.

We know... $\boxed{P \overset{val}{\models} Q \text{ if, and only if, } P \Rightarrow Q \text{ is a tautology.}}$

First, we establish that $\neg(P \Rightarrow Q) \overset{val}{\models} \neg Q \wedge (P \vee R)$:

$$
\begin{array}{ll}
 & \neg(P \Rightarrow Q) \\
\overset{val}{=} & \{ \text{Implication} \} \\
 & \neg(\neg P \vee Q) \\
\overset{val}{=} & \{ \text{De Morgan} \} \\
 & \neg\neg P \wedge \neg Q \\
\overset{val}{=} & \{ \text{Double Negation} \} \\
 & P \wedge \neg Q \\
\overset{val}{\models} & \{ \wedge\text{-}\vee\text{-weakening + Monotonicity} \} \\
 & \neg Q \wedge (P \vee \underline{R})
\end{array}
$$

Implication:
$P \Rightarrow Q \overset{val}{=} \neg P \vee Q$

De Morgan:
$\neg(P \wedge Q) \overset{val}{=} \neg P \vee \neg Q$
$\neg(P \vee Q) \overset{val}{=} \neg P \wedge \neg Q$

Double Negation:
$\neg\neg P \overset{val}{=} P$

Monotonicity:
(1) If $P \overset{val}{\models} Q$, then $P \wedge R \overset{val}{\models} Q \wedge R$
(2) If $P \overset{val}{\models} Q$, then $P \vee R \overset{val}{\models} Q \vee R$

Valid Argument

- An argument where conclusion is true if the propositions are true

"If Tweety is a bird then Tweety flies"
"Tweety is a bird"
"Tweety flies"

Premises
Conclusion

Valid Argument

- An argument where conclusion is true if the propositions are true

- Premise (axiom): a proposition held to be true

- Various notations (e.g. 1-3) for proofs of propositional logic formulas
  - (e.g. 2) Can derive the formula(s) below the line from comma-separated formulas above

1
$$\begin{array}{l} \quad p \rightarrow q \\ \quad p \\ \therefore q \end{array}$$

2
$$\frac{p \rightarrow q\ ,\ p}{q}$$

3
$$\begin{array}{l} p \rightarrow q \\ p \\ \hline q \end{array}$$

"If Tweety is a bird then Tweety flies"
"Tweety is a bird"
} Premises

"Tweety flies" → Conclusion

Valid Argument

- An argument where conclusion is true if the propositions are true

- Premise (axiom): a proposition held to be true

- Various notations (e.g. 1-3) for proofs of propositional logic formulas
  - (e.g. 2) Can derive the formula(s) below the line from comma-separated formulas above

- A series of statements form a valid argument if and only if `the conjunction (∧) of premises implying the conclusion' is a tautology (see next slide)

"If Tweety is a bird then Tweety flies"
"Tweety is a bird"
Premises

"Tweety flies" → Conclusion

1

$$\begin{array}{l} p \rightarrow q \\ p \\ \therefore q \end{array}$$

2

$$\frac{p \rightarrow q \,,\, p}{q}$$

3

$$\begin{array}{l} p \rightarrow q \\ \underline{p} \\ q \end{array}$$

Valid Argument Template

- Critical rows are rows with all premises true (satisfied)
  - Only need to calculate these in the truth table
- If in all critical rows the conclusion is true, then the argument is valid
  - Otherwise it is invalid
- Valid argument: `the conjunction (∧) of premises implying the conclusion' is a tautology
  - $(p \rightarrow q) \land p \rightarrow q$ in this example

$$\left.\begin{array}{l} p \rightarrow q \\ p \end{array}\right\} \text{Premises}$$

$$\therefore q \longrightarrow \text{Conclusion}$$

Valid Argument Template

- Critical rows are rows with all premises true (satisfied)
  - Only need to calculate these in the truth table
- If in all critical rows the conclusion is true, then the argument is valid
  - Otherwise it is invalid
- Valid argument: `the conjunction (∧) of premises implying the conclusion' is a tautology
  - $(p \rightarrow q) \wedge p \rightarrow q$ in this example

KB $\left\{ \begin{array}{l} p \rightarrow q \\ p \end{array} \right.$ — Premises

$\therefore q$ — Conclusion

$M(KB \cup \{f\}) = \{ p = T, q = T \}$

p	q	p→q	(p→q))∧p	((p→q))∧p) → q
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

No need to calculate

Valid Argument Template

- Critical rows are rows with all premises true (satisfied)
  - Only need to calculate these in the truth table
- If in all critical rows the conclusion is true, then the argument is valid
  - Otherwise it is invalid
- Valid argument: `the conjunction (∧) of premises implying the conclusion' is a tautology
  - $(p \rightarrow q) \wedge p \rightarrow q$ in this example

KB
$p \rightarrow q$
p
Premises
$\therefore q$
Conclusion

$M(KB \cup \{f\}) = \{ p = T, q = T \}$

p q	p→q	(p→q))∧p	((p→q))∧p) → q
T T	T	T	T
T F	F	F	T
F T	T	F	T
F F	T	F	T

No need to calculate

Invalid Argument Example

'if it is **f**alling *and* directly **a**bove me then I'll **r**un '

'It is **f**alling'

'it is *not* directly **a**bove me

premises

'I will *not* **r**un'

conclusion

$S=(f \land a \rightarrow r);$
$f;$
$\neg a;$
$\therefore \neg r$

premises

conclusion

a	r	f	$\neg$a	f $\land$ a	S	$\neg$r
T	T	T	F	T	T	F
T	T	F	F	F	T	F
T	F	T	F	T	F	T
T	F	F	F	F	T	T
F	T	T	T	F	T	F
F	T	F	T	F	T	F
F	F	T	T	F	T	T
F	F	F	T	F	T	T

Critical rows

Invalid argument : conclusion on 5^{th} row is false!

Counter Example

- A critical row with a false conclusion is a counter example that invalidates the argument

Role of Entailment in Valid Arguments

- Another way of looking at an argument being valid

- A valid logical argument is one in which the conclusion is entailed by the premises

Role of Entailment in Valid Arguments

- Another way of looking at an argument being valid

- A valid logical argument is one in which the conclusion is entailed by the premises

$$
\begin{array}{ll}
 & P \land Q \\
 & P \\
\therefore & P \lor Q
\end{array}
$$

P	Q	$P \land Q$	P	$P \lor Q$
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Critical row

$P \land Q \models P \lor Q$ ✓

Valid Argument!

Fallacy

- An error in reasoning that results in an invalid argument

Fallacy 1: converse error.

p	q	$p \to q$
F	T	T

$$p \to q;$$
$$q;$$
$$\therefore\ p$$

Example

- If it is Christmas, then it is a holiday.
- It is a holiday. Therefore, it is Christmas!

Fallacy

- An error in reasoning that results in an invalid argument

Fallacy 1: converse error.

p	q	$p \to q$
F	T	T

$$\begin{array}{l} p \to q; \\ \quad q; \\ \therefore\ p \end{array}$$

Example

- If it is Christmas, then it is a holiday.
- It is a holiday. Therefore, it is Christmas!

Fallacy 2: inverse error.

p	q	$p \to q$	$\neg p$	$\neg q$
F	T	T	T	F

$$\begin{array}{l} p \to q; \\ \neg p; \\ \therefore\ \neg q \end{array}$$

Example

- If it is raining, then I will stay at home.
- It is not raining. Therefore I would not stay at home!

Invalid Argument But Correct Conclusion

- An argument may be invalid, but it may still draw a correct conclusion (e.g. by coincidence)

- Example
  - If New York is a big city, then New York has tall buildings
  - New York has tall buildings
    - Therefore: New York is a big city

Invalid Argument But Correct Conclusion

- An argument may be invalid, but it may still draw a correct conclusion (e.g. by coincidence)

- Example
  - If New York is a big city, then New York has tall buildings
  - New York has tall buildings
    - Therefore: New York is a big city

- So what happened?
  - We have just made an invalid argument
    - Converse error!
  - But conclusion is true (a fact true by itself)

Inference Rules

- A rule of inference is a logical construct (i.e. argument) which takes premises, analyzes their syntax and returns a conclusion
  - Simple valid arguments that can be used as building blocks to construct more complicated valid arguments, or to derive a proof

Inference Rules: *Modus Ponens*

- If p, and p implies q, then q
  - The most famous form of syllogism
  - We have seen this many times already (in this course)

- Example:
  - "If it is snowing, then I will study."
  - "It is snowing."
  - "Therefore, I will study"

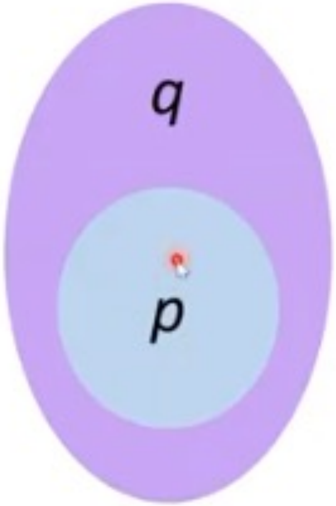
$$\begin{array}{l} \quad p \rightarrow q \\ \quad p \\ \therefore q \end{array}$$

Inference Rules: *Modus Ponens*

- If p, and p implies q, then q
  - The most famous form of syllogism
  - We have seen this many times already (in this course)

- Example:
  - “If it is snowing, then I will study.”
  - “It is snowing.”
  - “Therefore, I will study”

$$\begin{array}{l} \quad p \rightarrow q \\ \quad p \\ \therefore q \end{array}$$



Inference Rules: *Modus Ponens*

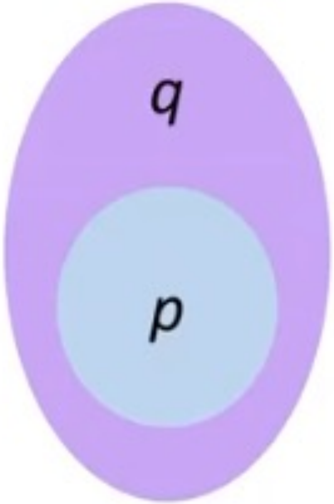
- If p, and p implies q, then q
  - The most famous form of syllogism
  - We have seen this many times already (in this course)

- Example:
  - "If it is snowing, then I will study."
  - "It is snowing."
  - "Therefore, I will study"

$$\begin{array}{l} \quad p \rightarrow q \\ \quad p \\ \therefore q \end{array}$$

Corresponding Tautology: $(p \land (p \rightarrow q)) \rightarrow q$

p	q	$p \rightarrow q$	$p \land (p \rightarrow q)$	$(p \land (p \rightarrow q)) \rightarrow q$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1



Inference Rules: *Modus Tollens*

- If not q and p implies q, then not p

- Example:
  - "If it is snowing, then I will study."
  - "It will not study."
  - "Therefore, it is not snowing"

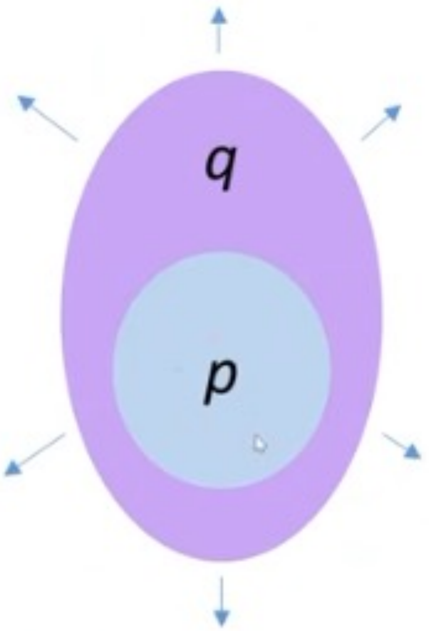
$$
\begin{array}{l}
\quad p \rightarrow q \\
\quad \neg q \\
\therefore \neg p
\end{array}
$$

Inference Rules: *Modus Tollens*

- If not q and p implies q, then not p

- Example:
  - "If it is snowing, then I will study."
  - "It will not study."
  - "Therefore, it is not snowing"

$$\begin{array}{ll} & p \rightarrow q \\ & \neg q \\ \therefore & \neg p \end{array}$$



Inference Rules: *Modus Tollens*

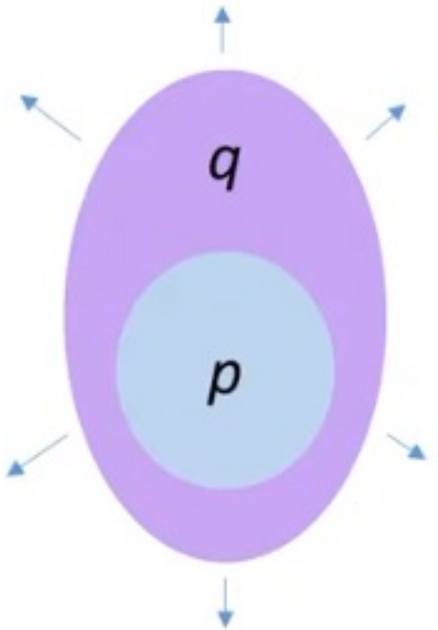
- If not q and p implies q, then not p

- Example:
  - "If it is snowing, then I will study."
  - "It will not study."
  - "Therefore, it is not snowing"

$$\begin{array}{l} \quad p \rightarrow q \\ \quad \neg q \\ \therefore \neg p \end{array}$$

Corresponding Tautology: $(\neg q \land (p \rightarrow q)) \rightarrow \neg p$

p	q	¬q	p → q	¬q ∧ (p → q)	¬p	(¬q ∧ (p → q)) → ¬p
0	0	1	1	1	1	1
0	1	0	1	0	1	1
1	0	1	0	0	0	1
1	1	0	1	0	0	1



Inference Rules: Resolution

- If p or q, and not p or r, then q or r
- Example:
  - "It is sunny or hot."
  - "It is not hot or it is dry."
  - "Therefore, it is hot or dry"

$$\begin{array}{rl} & p \vee q \\ & \neg p \vee r \\ \therefore & q \vee r \end{array}$$

Inference Rules: Resolution

- If p or q, and not p or r, then q or r
- Example:
  - "It is sunny or hot."
  - "It is not hot or it is dry."
  - "Therefore, it is hot or dry"

$$\begin{array}{rl} & p \lor q \\ & \neg p \lor r \\ \therefore & q \lor r \end{array}$$

Corresponding Tautology: $(p \lor q) \land (\neg p \lor r) \rightarrow q \lor r$

p	q	r	$\neg p$	$\neg p \lor r$	$p \lor q$	$(p \lor q) \land (\neg p \lor r)$	$q \lor r$	$(p \lor q) \land (\neg p \lor r) \rightarrow q \lor r$
0	0	0	1	1	0	0	0	1
0	0	1	1	1	0	0	1	1
0	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	0	0	1
1	0	1	0	1	1	1	1	1
1	1	0	0	0	1	0	1	1
1	1	1	0	1	1	1	1	1

Inference Rules: Hypothetical Syllogism

- Aka: Transitivity
- If p implies q, and q implies r, then p implies r
- Example:
  - "It is hot when it is sunny."
  - "It is dry when it is hot."
  - "Therefore, it must be dry when it is sunny"

$$\begin{array}{l} \quad p \rightarrow q \\ \quad q \rightarrow r \\ \hline \therefore p \rightarrow r \end{array}$$

Inference Rules: Hypothetical Syllogism

- Aka: Transitivity
- If p implies q, and q implies r, then p implies r
- Example:
  - "It is hot when it is sunny."
  - "It is dry when it is hot."
  - "Therefore, it must be dry when it is sunny"

$$\begin{array}{r} p \rightarrow q \\ q \rightarrow r \\ \therefore\ p \rightarrow r \end{array}$$

Corresponding Tautology: $(p \rightarrow q) \land (q \rightarrow r) \rightarrow (p \rightarrow r)$

p	q	r	$p \rightarrow q$	$q \rightarrow r$	$(p \rightarrow q) \land (q \rightarrow r)$	$p \rightarrow r$	$(p \rightarrow q) \land (q \rightarrow r) \rightarrow (p \rightarrow r)$
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	1

Inference Rules: Disjunctive Syllogism

- Aka: Elimination
- If p or q, and not p, then q

- Example:
  - "It is sunny or hot."
  - "It is not sunny."
  - "Therefore, its hot."

$$
\begin{array}{l}
p \lor q \\
\neg p \\
\therefore q
\end{array}
$$

Inference Rules: Disjunctive Syllogism

- Aka: Elimination
- If p or q, and not p, then q

- Example:
  - "It is sunny or hot."
  - "It is not sunny."
  - "Therefore, its hot."

$$\begin{array}{ll} & p \lor q \\ & \neg p \\ \therefore & q \end{array}$$

Corresponding Tautology: $(p \lor q) \land (\neg p) \rightarrow q$

p	q	$p \lor q$	$\neg p$	$(p \lor q) \land (\neg p)$	$(p \lor q) \land (\neg p) \rightarrow q$
0	0	0	1	0	1
0	1	1	1	1	1
1	0	1	0	0	1
1	1	1	0	0	1

IR: Simplification and Addition

- Simplification: (particularizing)
  - If p and q then p
  - Corresponding Tautology: $p \wedge q \rightarrow p$

$$\begin{array}{l} \quad p \wedge q \\ \therefore p \end{array}$$

IR: Simplification and Addition

- Simplification: (particularizing)
  - If p and q then p
  - Corresponding Tautology: $p \land q \to p$

$$\begin{array}{l} \quad p \land q \\ \therefore p \end{array}$$

p	q	$p \land q$	$p \land q \to p$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Simplification

IR: Simplification and Addition

- Simplification: (particularizing)
  - If p and q then p
  - Corresponding Tautology: $p \land q \rightarrow p$

$$\begin{array}{l} \quad p \land q \\ \therefore p \end{array}$$

- Addition: (generalization)
  - If p then p or q
  - Corresponding Tautology: $p \rightarrow p \lor q$

$$\begin{array}{l} \quad p \\ \therefore p \lor q \end{array}$$

p	**q**	**$p \land q$**	**$p \land q \rightarrow p$**
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Simplification

IR: Simplification and Addition

- Simplification: (particularizing)
  - If p and q then p
  - Corresponding Tautology: $p \land q \rightarrow p$

$$\begin{array}{l} \quad p \land q \\ \therefore p \end{array}$$

- Addition: (generalization)
  - If p then p or q
  - Corresponding Tautology: $p \rightarrow p \lor q$

$$\begin{array}{l} \quad p \\ \therefore p \lor q \end{array}$$

p	**q**	**$p \land q$**	**$p \land q \rightarrow p$**
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Simplification

p	**q**	**$p \lor q$**	**$p \rightarrow p \lor q$**
0	0	0	1
0	1	1	1
1	0	1	1
1	1	1	1

Addition

IR: Conjunction and Contradiction

- Conjunction: (conjunctive addition)
  - If p and if q then p and q
  - Corresponding Tautology: $p \land q \rightarrow p \land q$

$$
\begin{array}{l}
\quad p \\
\quad q \\
\therefore p \land q
\end{array}
$$

IR: Conjunction and Contradiction

- Conjunction: (conjunctive addition)
  - If p and if q then p and q
  - Corresponding Tautology: $p \wedge q \rightarrow p \wedge q$

$$\begin{array}{l} \quad p \\ \quad q \\ \therefore p \wedge q \end{array}$$

- Contradiction:
  - If not p implies FALSE then p
  - Corresponding Tautology: $\neg p \rightarrow F \rightarrow p$

$$\begin{array}{l} \quad \neg p \rightarrow F \\ \therefore p \end{array}$$

p	**q**	**$p \rightarrow q$**	**$p \wedge q \rightarrow p \wedge q$**
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	1

Conjunction

IR: Conjunction and Contradiction

- Conjunction: (conjunctive addition)
  - If p and if q then p and q
  - Corresponding Tautology: $p \land q \rightarrow p \land q$

$$\begin{array}{l} \quad p \\ \quad q \\ \therefore p \land q \end{array}$$

- Contradiction:
  - If not p implies FALSE then p
  - Corresponding Tautology: $\neg p \rightarrow F \rightarrow p$

$$\begin{array}{l} \quad \neg p \rightarrow F \\ \therefore p \end{array}$$

p	q	$p \rightarrow q$	$p \land q \rightarrow p \land q$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	1

Conjunction

p	q	$\neg p$	F	$\neg p \rightarrow F$	$\neg p \rightarrow F \rightarrow p$
0	0	1	0	0	1
0	1	1	0	0	1
1	0	0	0	1	1
1	1	0	0	1	1

Contradiction

Summary: Inference Rules

Modus Ponens

$$\begin{array}{l} p \rightarrow q \\ p \\ \therefore q \end{array}$$

Modus Tollens

$$\begin{array}{l} p \rightarrow q \\ \neg q \\ \therefore \neg p \end{array}$$

Resolution

$$\begin{array}{l} p \vee q \\ \neg p \vee r \\ \therefore q \vee r \end{array}$$

Simplification

$$\begin{array}{l} p \wedge q \\ \therefore p \end{array}$$

Addition

$$\begin{array}{l} p \\ \therefore p \vee q \end{array}$$

Disjunctive Syllogism

$$\begin{array}{l} p \vee q \\ \neg p \\ \therefore q \end{array}$$

Hypothetical Syllogism

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \therefore p \rightarrow r \end{array}$$

Conjunction

$$\begin{array}{l} p \\ q \\ \therefore p \wedge q \end{array}$$

Contradiction

$$\begin{array}{l} \neg p \rightarrow F \\ \therefore p \end{array}$$

Deductive Reasoning from a Knowledge Base

Deductive Reasoning

- In propositional logic, deductive reasoning takes the form of demonstrating entailment between a knowledge base and some conclusion

Logic: another thing that penguins aren't very good at.

KB Satisfiability and Entailment

- Knowledge-base (**KB**) is satisfiable if $M(KB) \neq \emptyset$
  - The union of satisfied models over all propositions in KB is not an empty set

  - Otherwise the KB contradicts the conclusion
    - Assume conclusion = proposition ‘p’

- **KB** entails the conclusion ‘p’ (i.e. $KB \models p$) if

$$M(KB \cup \{p\}) = M(KB)$$

  - i.e. in every world where KB is true, p is also true

Demonstrating KB Entailment

- In practice how do we go about this?

- What challenges do we face when working with a large knowledge base and/or complex propositions?

Demonstrating KB Entailment

- In practice how do we go about this?

- What challenges do we face when working with a large knowledge base and/or complex propositions?

- Methods for demonstrating KB entailment:

  - Truth Table Checking (Satisfiability)

  - Proof by Contradiction/Refutation
    - Truth Table Checking Unsatisfiability
    - Theorem Proving: Determining Unsatisfiability

  - Resolution (Uses proof by contradiction)

KB Entailment: Truth Table Method

- Model checking: enumerating models and showing that the sentence must hold in all models

- Method for computing whether a set of premises logically entails a conclusion

  1. Form a truth table for the propositional constants occurring in the premises and conclusion; add a column for the premises and a column for the conclusion

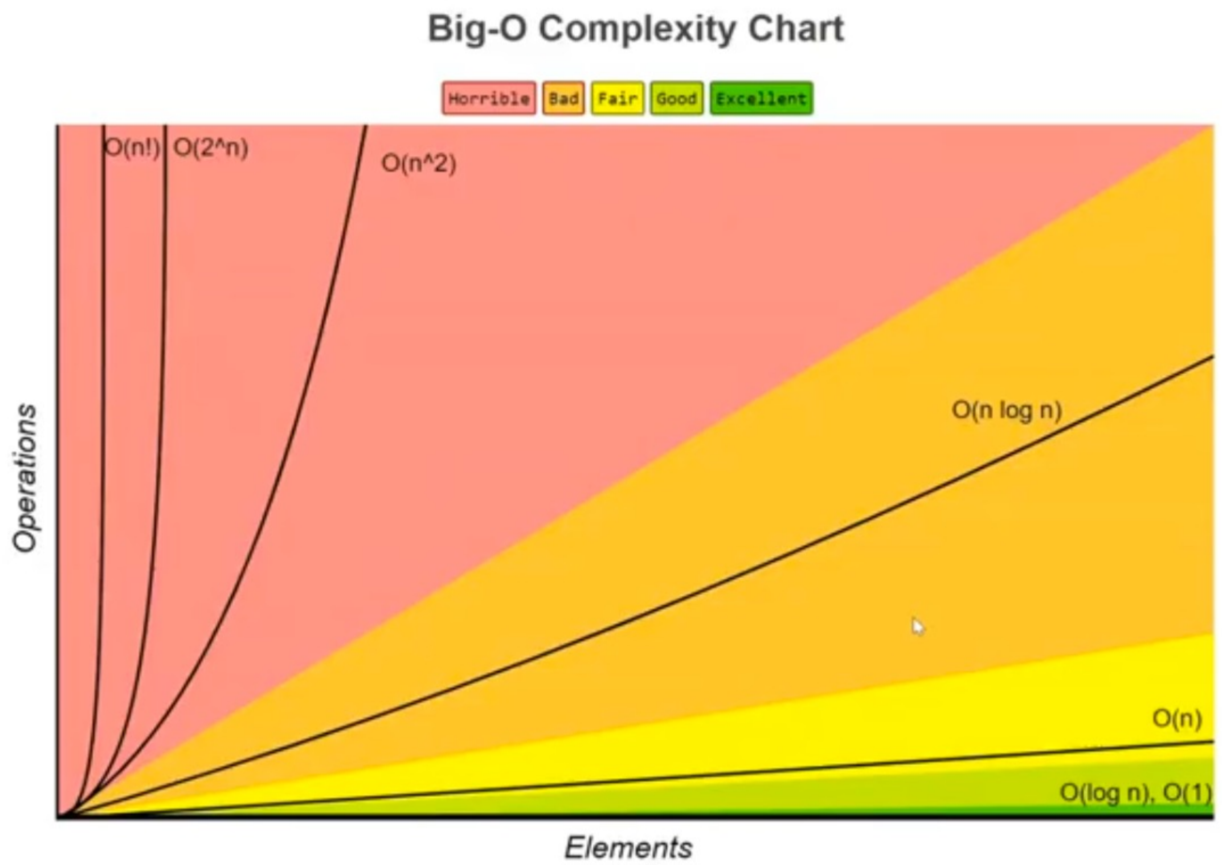
  2. Evaluate the premises for each row in the table

  3. Evaluate the conclusion for each row in the table

  4. If every row that satisfies the premises also satisfies the conclusion, then the premises logically entail the conclusion ✔

Truth Table Method Disadvantage

- Given *n* propositional constants it can take 2^n time to perform the inference with a truth table
  - Time complexity of this algorithm would be $O(2^n)$
  - Could be infeasible for even $n = 100$



KB Entailment: Proof by Contradiction

A.k.a Proof by Refutation

KB $\models$ q if and only if KB $\land \neg$q is unsatisfiable

- Where does this come from?
  - Deduction theorem $P \overset{val}{\models} Q$ if, and only if, $P \Rightarrow Q$ is a tautology.
  - Thus we know that $\neg$(KB $\longrightarrow$ q) is unsatisfiable:
    - Negation of a tautology

KB Entailment: Proof by Contradiction

A.k.a Proof by Refutation

KB $\models$ q if and only if KB $\land \neg$q is unsatisfiable

- Where does this come from?
  - Deduction theorem $P \overset{val}{\models} Q$ if, and only if, $P \Rightarrow Q$ is a tautology.
  - Thus we know that $\neg$(KB $\rightarrow$ q) is unsatisfiable:
    - Negation of a tautology
  - Derive by equivalence:
    - $\neg(\text{KB} \rightarrow q) \quad \equiv \quad \neg(\neg\text{KB} \lor q) \quad \equiv \quad \text{KB} \land \neg q$

Implication:
$P \Rightarrow Q \overset{val}{=\!=} \neg P \lor Q$

De Morgan:
$\neg(P \land Q) \overset{val}{=\!=} \neg P \lor \neg Q,$
$\neg(P \lor Q) \overset{val}{=\!=} \neg P \land \neg Q$

KB Entailment: Proof by Contradiction

A.k.a Proof by Refutation

KB $\models$ q if and only if KB ∧ ¬q is unsatisfiable

- Where does this come from?
  - Deduction theorem — $P \overset{val}{\models} Q$ if, and only if, $P \Rightarrow Q$ is a tautology.
  - Thus we know that ¬(KB → q) is unsatisfiable:
    - Negation of a tautology
  - Derive by equivalence:
    - ¬(KB → q) ≡ ¬(¬KB ∨ q) ≡ KB ∧ ¬q

Implication:

$$P \Rightarrow Q \overset{val}{=\!=} \neg P \vee Q$$

De Morgan:

$$\neg(P \wedge Q) \overset{val}{=\!=} \neg P \vee \neg Q,$$
$$\neg(P \vee Q) \overset{val}{=\!=} \neg P \wedge \neg Q$$

Demonstrating Unsatisfiability

KB $\models$ q if and only if KB $\wedge \neg$q is unsatisfiable

- Add ¬q to KB and derive a contradiction (unsatisfiable)

Demonstrating Unsatisfiability

KB $\models$ q if and only if KB $\wedge \neg$q is unsatisfiable

- Add ¬q to KB and derive a contradiction (unsatisfiable)

- Method for computing whether a set of premises logically entails a conclusion

1. Form a truth table for the propositional constants.
2. For each sentence in the set and each row in the truth table, check whether the row satisfies the sentence. If not, cross out the row.
3. If all rows are crossed off then KB $\models$ q ✔

Demonstrating Unsatisfiability

KB $\models$ q if and only if KB $\wedge \neg$q is unsatisfiable

- Add ¬q to KB and derive a contradiction (unsatisfiable)

- Method for computing whether a set of premises logically entails a conclusion

  1. Form a truth table for the propositional constants.
  2. For each sentence in the set and each row in the truth table, check whether the row satisfies the sentence. If not, cross out the row.
  3. If all rows are crossed off then KB $\models$ q ✔

- Determining satisfiability of sentences in propositional logic was the first problem proved to be NP-complete

Complexity Classes in Problem Solving

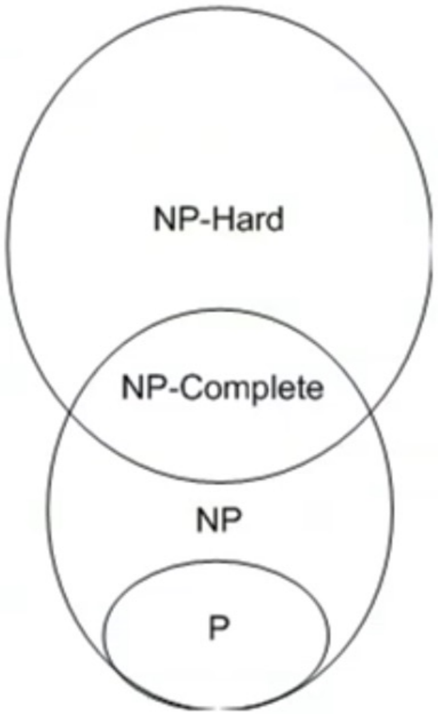
- Alluding to challenges in search for decisions, conclusions, solutions

- **P**: can be solved on deterministic Turing machine in polynomial time $T(n) = O(n^k)$

- **NP**: can be solved on non-deterministic Turing machine in polynomial time

- **NP-Complete**: can be solved by restricted class of brute force algorithms in polynomial time

- **NP-Hard**: non-deterministic polynomial-time hardness
  - At least as hard as the hardest problems in NP



Assignments

- Reminder: Assignment 2

- Goals:
    - Propositional Logic & First Order Logic
        - Practice working with the syntax and semantics of logic
        - Practice manipulating logical expressions in an 'algebraic' manner
        - Apply logic to making inferences and determining 'truth'
        - Understand two of the simplest inferences approaches that we will focus on primarily in this course:
            - Forward chaining
            - Backward chaining

More on Demonstrating Unsatisfiability

- Can we avoid the truth table? Yes!

More on Demonstrating Unsatisfiability

- Can we avoid the truth table? Yes!

- Theorem Proving: applying rules of inference directly to the sentences in our knowledge base to construct a proof of the desired sentence without consulting models

- This can be done by hand (next slide), or we could apply a search algorithm to find a sequence of steps that constitutes a proof

- $p \equiv (q \vee r), \neg p \quad \models \quad \neg q \quad ?$

KB Entailment: Theorem Proving

$p \equiv (q \lor r), \neg p \models \neg q$

Using a mix of equivalences and inference rules

KB Entailment: Theorem Proving

$p \equiv (q \lor r), \neg p \models \neg q$

Using a mix of equivalences and inference rules

- $p \equiv (q \lor r)$
- $(p \rightarrow (q \lor r)) \land ((q \lor r) \rightarrow p)$
- $(q \lor r) \rightarrow p$
- $\neg p \rightarrow \neg(q \lor r)$
- $\neg(q \lor r)$
- $\neg q \land \neg r$

Entailment Proved ✔

Bi-implication:
$P \Leftrightarrow Q \overset{val}{=\!=} (P \Rightarrow Q) \land (Q \Rightarrow P)$

Simplification

$p \land q$
$\therefore p$

Contraposition:
$P \Rightarrow Q \overset{val}{=\!=} \neg Q \Rightarrow \neg P$

$\neg p$

Modus Ponens

$p \rightarrow q$
p
$\therefore q$

De Morgan:
$\neg(P \land Q) \overset{val}{=\!=} \neg P \lor \neg Q,$
$\neg(P \lor Q) \overset{val}{=\!=} \neg P \land \neg Q$

KB Entailment: Resolution

- Inference algorithms searching for proofs must be concerned with completeness
  - i.e. Are available inference rules adequate to derive the proof

KB Entailment: Resolution

- Inference algorithms searching for proofs must be concerned with completeness
  - i.e. Are available inference rules adequate to derive the proof
- Alternatively there is a single inference rule (resolution rule):
  - Yields a complete inference algorithm when coupled with a complete search algorithm

$$\begin{array}{rl} & a \lor b \\ & \neg b \lor c \\ \therefore & a \lor c \end{array} \quad \overset{val}{\equiv} \quad \begin{array}{rl} & a \lor b \\ & b \rightarrow c \\ \therefore & a \lor c \end{array}$$

Resolution Rule

Generalization of Modus Ponens

KB Entailment: Resolution

- Inference algorithms searching for proofs must be concerned with completeness
  - i.e. Are available inference rules adequate to derive the proof
- Alternatively there is a single inference rule (resolution rule):
  - Yields a complete inference algorithm when coupled with a complete search algorithm
- Generalized resolution rule:
  - Allows clauses with an arbitrary number of literals

$$\begin{array}{r} a \vee b \\ \neg b \vee c \\ \hline \therefore\ a \vee c \end{array} \quad \overset{val}{\equiv} \quad \begin{array}{r} a \vee b \\ b \rightarrow c \\ \hline \therefore\ a \vee c \end{array} \qquad \begin{array}{c} (a_1 \vee \ldots \vee a_m \vee b) \\ (\neg b \vee c_1 \vee \ldots \vee c_n) \\ \therefore (a_1 \vee \ldots \vee a_m \vee c_1 \vee \ldots \vee c_n) \end{array}$$

Resolution Rule | Generalization of Modus Ponens | Generalized Resolution Rule

Conjuctive Normal Form (CNF)

- To keep automatic proof systems as simple as possible, these are usually made to operate on formulas in conjunctive normal form

Conjuctive Normal Form (CNF)

- To keep automatic proof systems as simple as possible, these are usually made to operate on formulas in conjunctive normal form

- A formula is in CNF if and only if it consists of a conjunction of clauses $(K_1 \land K_2 \land \ldots \land K_m)$
- A clause K_i consists of a disjunction of literals $(L_1 \lor L_2 \lor \ldots \lor L_m)$
- A literal is a positive or negative proposition

Conjuctive Normal Form (CNF)

- To keep automatic proof systems as simple as possible, these are usually made to operate on formulas in conjunctive normal form

- A formula is in CNF if and only if it consists of a conjunction of clauses $(K_1 \land K_2 \land \ldots \land K_m)$

- A clause K_i consists of a disjunction of literals $(L_1 \lor L_2 \lor \ldots \lor L_m)$

- A literal is a positive or negative proposition

- Example: $(a \lor b \lor \neg c) \land (a \lor b) \land (\neg b \lor \neg c)$

- Every propositional logic formula can be transformed into an equivalent CNF

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \lor b \to c \land d$

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \vee b \longrightarrow c \wedge d$

Implication:
$P \Rightarrow Q \overset{val}{=\!=} \neg P \vee Q$

$\neg(a \vee b) \vee (c \wedge d)$

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \vee b \rightarrow c \wedge d$

$\neg(a \vee b) \vee (c \wedge d)$

$(\neg a \wedge \neg b) \vee (c \wedge d)$

Implication:

$$P \Rightarrow Q \overset{val}{=\!=} \neg P \vee Q$$

De Morgan:

$$\neg(P \wedge Q) \overset{val}{=\!=} \neg P \vee \neg Q,$$

$$\neg(P \vee Q) \overset{val}{=\!=} \neg P \wedge \neg Q$$

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \vee b \longrightarrow c \wedge d$

$\neg(a \vee b) \vee (c \wedge d)$

$(\neg a \wedge \neg b) \vee (c \wedge d)$

$(\neg a \vee (c \wedge d)) \wedge (\neg b \vee (c \wedge d))$

Implication:

$$P \Rightarrow Q \stackrel{val}{=\!=} \neg P \vee Q$$

De Morgan:

$$\neg(P \wedge Q) \stackrel{val}{=\!=} \neg P \vee \neg Q,$$
$$\neg(P \vee Q) \stackrel{val}{=\!=} \neg P \wedge \neg Q$$

Distributivity:

$$P \wedge (Q \vee R) \stackrel{val}{=\!=} (P \wedge Q) \vee (P \wedge R),$$
$$P \vee (Q \wedge R) \stackrel{val}{=\!=} (P \vee Q) \wedge (P \vee R)$$

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \lor b \rightarrow c \land d$

$\neg(a \lor b) \lor (c \land d)$

$(\neg a \land \neg b) \lor (c \land d)$

$(\neg a \lor (c \land d)) \land (\neg b \lor (c \land d))$

$((\neg a \lor c) \land (\neg a \lor d)) \land ((\neg b \lor c) \land (\neg b \lor d))$

Implication:
$P \Rightarrow Q \stackrel{val}{=\!=} \neg P \lor Q$

De Morgan:
$\neg(P \land Q) \stackrel{val}{=\!=} \neg P \lor \neg Q,$
$\neg(P \lor Q) \stackrel{val}{=\!=} \neg P \land \neg Q$

Distributivity:
$P \land (Q \lor R) \stackrel{val}{=\!=} (P \land Q) \lor (P \land R),$
$P \lor (Q \land R) \stackrel{val}{=\!=} (P \lor Q) \land (P \lor R)$

Proposition to CNF

- Convert proposition to Conjunctive Normal Form

Example: $a \lor b \rightarrow c \land d$

$\neg(a \lor b) \lor (c \land d)$

$(\neg a \land \neg b) \lor (c \land d)$

$(\neg a \lor (c \land d)) \land (\neg b \lor (c \land d))$

$((\neg a \lor c) \land (\neg a \lor d)) \land ((\neg b \lor c) \land (\neg b \lor d))$

$(\neg a \lor c) \land (\neg a \lor d) \land (\neg b \lor c) \land (\neg b \lor d)$

Implication:

$$P \Rightarrow Q \stackrel{val}{=\!=} \neg P \lor Q$$

De Morgan:

$$\neg(P \land Q) \stackrel{val}{=\!=} \neg P \lor \neg Q,$$
$$\neg(P \lor Q) \stackrel{val}{=\!=} \neg P \land \neg Q$$

Distributivity:

$$P \land (Q \lor R) \stackrel{val}{=\!=} (P \land Q) \lor (P \land R),$$
$$P \lor (Q \land R) \stackrel{val}{=\!=} (P \lor Q) \land (P \lor R)$$

Associativity:

$$(P \land Q) \land R \stackrel{val}{=\!=} P \land (Q \land R),$$
$$(P \lor Q) \lor R \stackrel{val}{=\!=} P \lor (Q \lor R),$$

Simplifying CNF to 'Pure' CNF

- An empty clause is false (F)
  - No options to satisfy
  - This makes the sentence False, as it is an impossible requirement

$$(\ ) \land (\neg a \lor d) \land (\neg b \lor c) \equiv F$$

Simplifying CNF to 'Pure' CNF

- An empty clause is false (F)
  - No options to satisfy
  - This makes the sentence False, as it is an impossible requirement

$$(\,) \land (\neg a \lor d) \land (\neg b \lor c) \quad \equiv \quad F$$

- A sentence with no clauses is True
  - No requirements

$$(\,) \quad \equiv \quad T$$

Simplifying CNF to 'Pure' CNF

- An empty clause is false (F)
  - No options to satisfy
  - This makes the sentence False, as it is an impossible requirement

$$(\ ) \land (\neg a \lor d) \land (\neg b \lor c) \equiv F$$

- A sentence with no clauses is True
  - No requirements

$$(\ ) \equiv T$$

- Valid clauses can be deleted as soon as they appear

$$(\neg a \lor d \lor a) \land (\neg b \lor c) \equiv (\neg b \lor c)$$

Simplifying CNF to 'Pure' CNF

- An empty clause is false (F)
  - No options to satisfy
  - This makes the sentence False, as it is an impossible requirement

$$(\,) \land (\neg a \lor d) \land (\neg b \lor c) \equiv F$$

- A sentence with no clauses is True
  - No requirements

$$(\,) \equiv T$$

- Valid clauses can be deleted as soon as they appear

$$(\neg a \lor d \lor a) \land (\neg b \lor c) \equiv (\neg b \lor c)$$

- Repeated literals in the same clause can be simplified
  - Factorization

$$(a \lor d \lor a) \land (\neg b \lor c) \equiv (a \lor d) \land (\neg b \lor c)$$

Simplifying CNF to 'Pure' CNF

- An empty clause is false (F)
  - No options to satisfy
  - This makes the sentence False, as it is an impossible requirement

$$( ) \land (\neg a \lor d) \land (\neg b \lor c) \equiv F$$

- A sentence with no clauses is True
  - No requirements

$$( ) \equiv T$$

- Valid clauses can be deleted as soon as they appear

$$(\neg a \lor d \lor a) \land (\neg b \lor c) \equiv (\neg b \lor c)$$

- Repeated literals in the same clause can be simplified
  - Factorization

$$(a \lor d \lor a) \land (\neg b \lor c) \equiv (a \lor d) \land (\neg b \lor c)$$

- If a clause c is included in another clause c' (c subsumes c'),then clause c' can be deleted

$$(a \lor d) \land (a \lor d \lor c) \equiv (a \lor d \lor c)$$

KB Entailment: Resolution Algorithm

- **Recall:** To demonstrate that KB entails q we can show that (KB ∧ ¬q) is unsatisfiable (proof by contradiction)

KB Entailment: Resolution Algorithm

- **Recall:** To demonstrate that KB entails q we can show that (KB ∧ ¬q) is unsatisfiable (proof by contradiction)

- Resolution algorithm:
  - All sentences in KB are converted to CNF and are conjunctively connected along with the conjecture (¬q)

KB Entailment: Resolution Algorithm

- **Recall:** To demonstrate that KB entails q we can show that (KB ∧ ¬q) is unsatisfiable (proof by contradiction)

- Resolution algorithm:
  - All sentences in KB are converted to CNF and are conjunctively connected along with the conjecture (¬q)
  - Resolution rule applied to each unique pair of clauses
    - New clauses are added to the total set of clauses

KB Entailment: Resolution Algorithm

- **Recall:** To demonstrate that KB entails q we can show that (KB ∧ ¬q) is unsatisfiable (proof by contradiction)

- Resolution algorithm:
  - All sentences in KB are converted to CNF and are conjunctively connected along with the conjecture (¬q)

  - Resolution rule applied to each unique pair of clauses
    - New clauses are added to the total set of clauses

  - Clauses that resolve to True can be discarded from further consideration

KB Entailment: Resolution Algorithm

- **Recall:** To demonstrate that KB entails q we can show that (KB ∧ ¬q) is unsatisfiable (proof by contradiction)

- Resolution algorithm:
  - All sentences in KB are converted to CNF and are conjunctively connected along with the conjecture (¬q)
  - Resolution rule applied to each unique pair of clauses
    - New clauses are added to the total set of clauses
  - Clauses that resolve to True can be discarded from further consideration
  - Process continues until either:
    1. Two clauses resolve to yield the empty clause, i.e. FALSE (KB entails q)
    2. No new clauses can be added (KB does not entail q)

Example of Resolution Algorithm

- KB: $(\neg c \vee d)$, $(\neg d \vee \neg q \vee c)$, $(q \vee d)$, $(\neg d)$
- CNF $\boxed{(\neg c \vee d) \wedge (\neg d \vee \neg q \vee c) \wedge (q \vee d) \wedge (\neg d) \wedge \neg q}$

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\boxed{(d \lor \neg q \lor \neg d)}$

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\boxed{(d \lor \neg q \lor \neg d)}$

TRUE

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\boxed{\cancel{(d \lor \neg q \lor \neg d)}}$ $\boxed{(\neg q \lor c \lor \neg c)}$

TRUE

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\boxed{(d \lor \neg q \lor \neg d)}$ $\boxed{(\neg q \lor c \lor \neg c)}$ $\boxed{(\neg d \lor c \lor d)}$

TRUE TRUE

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\cancel{(d \lor \neg q \lor \neg d)}$	$\cancel{(\neg q \lor c \lor \neg c)}$	$\cancel{(\neg d \lor c \lor d)}$	$(\neg q \lor c \lor q)$
TRUE	**TRUE**	**TRUE**	

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

~~$(d \lor \neg q \lor \neg d)$~~	~~$(\neg q \lor c \lor \neg c)$~~	~~$(\neg d \lor c \lor d)$~~	~~$(\neg q \lor c \lor q)$~~	$(\neg c)$
TRUE	**TRUE**	**TRUE**	**TRUE**	

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\cancel{(d \lor \neg q \lor \neg d)}$	$\cancel{(\neg q \lor c \lor \neg c)}$	$\cancel{(\neg d \lor c \lor d)}$	$\cancel{(\neg q \lor c \lor q)}$	$(\neg c)$	(q)
TRUE	**TRUE**	**TRUE**	**TRUE**		

Example of Resolution Algorithm

- KB: $(\neg c \lor d)$, $(\neg d \lor \neg q \lor c)$, $(q \lor d)$, $(\neg d)$

- CNF $\boxed{(\neg c \lor d) \land (\neg d \lor \neg q \lor c) \land (q \lor d) \land (\neg d) \land \neg q}$

$\boxed{\cancel{(d \lor \neg q \lor \neg d)}}$ $\boxed{\cancel{(\neg q \lor c \lor \neg c)}}$ $\boxed{\cancel{(\neg d \lor c \lor d)}}$ $\boxed{\cancel{(\neg q \lor c \lor q)}}$ $\boxed{(\neg c)}$ $\boxed{(q)}$

TRUE TRUE TRUE TRUE

In the set of resulting clauses do any resolve to an empty set?

$$\boxed{(q) \land \neg q} \xrightarrow{\text{Resolution Rule}} \boxed{(\ )}$$

(KB entails q) ✔

Example of Resolution Algorithm

$$S = \{p \overset{1}{\vee} q, p \overset{2}{\vee} r, \neg q \overset{3}{\vee} \neg r, \overset{4}{\neg p}\}$$

#	clause	from
5	$p \vee \neg r$	(1, 3)
6	q	(1, 4)
7	$p \vee \neg q$	(2, 3)
8	r	(2, 4)
9	p	(2, 5)
10	$\neg r$	(3, 6)
11	$\neg q$	(3, 8)
12	$\neg r$	(4, 5)
13	$\neg q$	(4, 7)
14	F	(4, 9)

Summary of Propositional Logic

1. A precisely defined formal knowledge representation
   - Propositions and Connectives

2. Propositional Truth

3. Rules for calculating with abstract propositions:
   - Equivalences
   - Inference Rules (multiple propositions)

4. Reasoning:
   - Entailment (a precise form of inference)
   - Arguments (combining multiple propositions to draw a conclusion)
   - Deductive reasoning proofs from a knowledge base