PROPOSITIONAL LOGIC 1

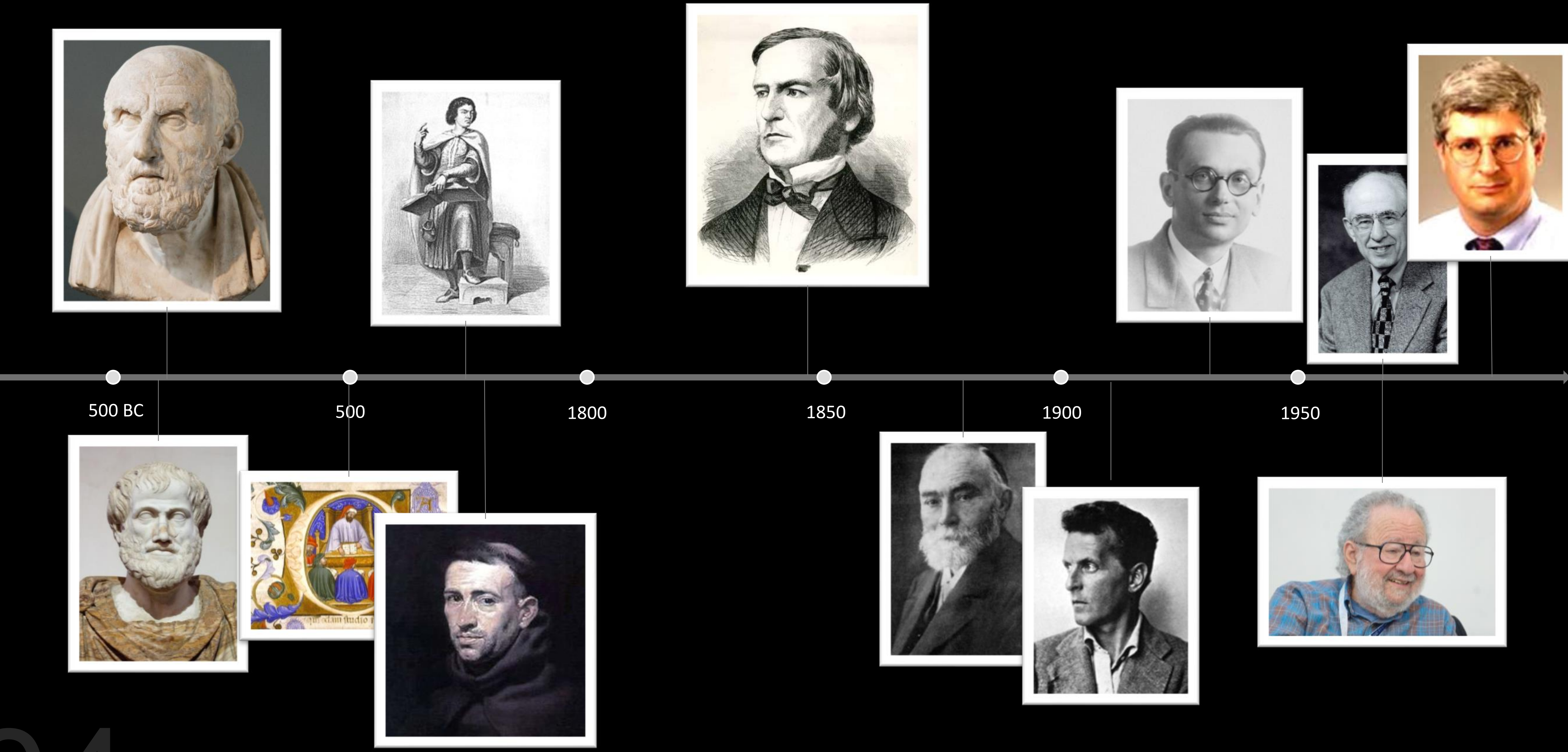
ARTIFICIAL INTELLIGENCE | COMP 131

TODAY ON AI

- What is Logic?
- Propositional Logic
- Questions?

SECTION 01

What is Logic?



04

Brief history of Logic

Logic is a way of formally representing the **state of the world** and the world's **rules of operation** so that we can make **rational decisions** and learn **new knowledge** based on our **existing one**.

It allows to:

- **Express knowledge** using a formal language
- **To carry out** reasoning in that language

There are several types of logic. Each type is increasingly complex as it captures more advanced concepts:

LANGUAGE	ONTOLOGICAL COMMITMENT	EPISTEMOLOGICAL COMMITMENT
Propositional logic	Facts	True / False / Unknown
First-order logic	Facts, Objects, Relations	True / False / Unknown
Temporal logic	Facts, Objects, Relations, Times	True / False / Unknown
Probability theory	Facts	Degree of belief ∈ [0, 1]
Fuzzy logic	Facts with degree of truth ∈ [0, 1]	Known interval value
Markov logic	Facts, Objects, Relations	Degree of belief ∈ [0, 1]

- **Programming languages**:
  - They are formal and not ambiguous
  - Unfortunately they lacks expressivity as they cannot accommodate partial information

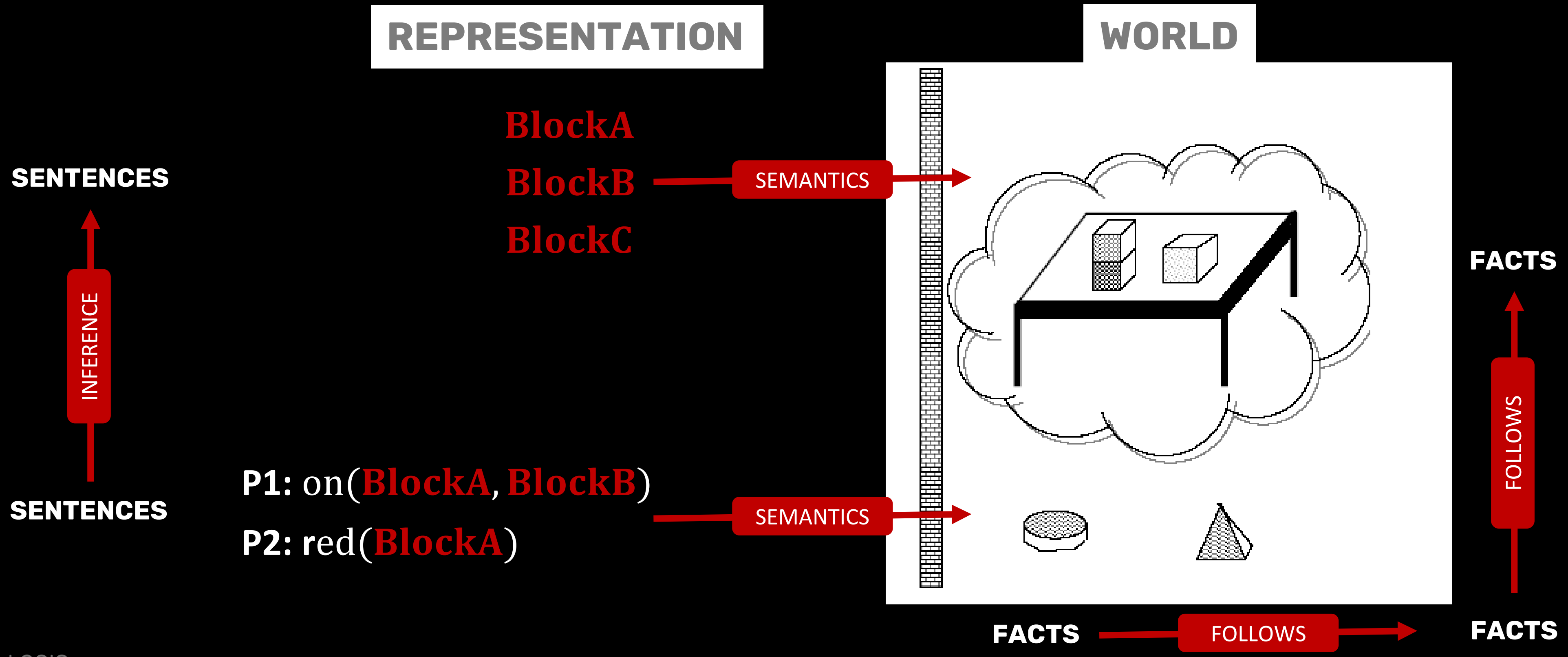
- **Natural Language**:
  - Very expressive but also ambiguous:
  - Inference possible, but hard to automate

Flying planes can be dangerous.
The teacher gave the boys an apple.

- A good **representation language** is:
  - Both formal and can express partial information
  - Can accommodate inference

The **fundamental elements** of a formal logic are:

- **Syntax**: it is the set of symbols and rules used to express knowledge
- **Semantics**: it specifies the way symbols and sentences relate to the world
- **Inference procedures**: they describe the rules for deriving new sentences (and therefore, new semantics) from existing sentences



SECTION 02

Propositional Logic

- A **sentence** expresses a possible condition of the world

- A sentence can either **true (T)** or **false (F)**

- Most basic sentences are called **simple**, **predicates** or **atomic sentences**
  am_wet, is_raining, have_umbrella

- **Predicates**: Symbols or constants **true (T)** or **false (F)**
- **Symbols**: p, q, s, …
- Sentences are combined by **connectives** to produce other **sentences**:

$\wedge$	**AND**	Conjunction
$\vee$	**OR**	Disjunction
$\rightarrow$	**IMPLIES**	Implication / conditional
$\leftrightarrow$	**IS EQUIVALENT**	Biconditional
$\neg$	**NOT**	Negation

A **sentence** is a well-formed formula that can be defined recursively:

- A **symbol** is a proposition
- If s is a sentence, then $\neg s$ is a sentence
- If s is a sentence, then (s) is a sentence
- If s and t are sentences, then $(s \wedge t), (s \vee t), (s \rightarrow t)$ and $(s \leftrightarrow t)$ are sentences
- A sentence results from a finite number of iterations of the above rules

- **Operator precedence**: $\neg \ \wedge \ \vee \ \rightarrow \ \leftrightarrow$

PROPOSITIONAL LOGIC GRAMMAR

Sentence	→	Predicate | Proposition
Proposition	→	(Sentence)
	|	Sentence Connective Sentence
	|	¬ Sentence
Predicate	→	**true (T)** | **false (F)** | *Symbol*
Connective	→	∧, ∨, →, ↔

- o means "**It is hot**"
- h means "**It is humid**"
- r means **"It is raining"**
- $(o \wedge h) \rightarrow \mathrm{r}$ means **"IF it is hot AND humid, THEN it is raining"**
- $q \rightarrow p$ means **"IF it is humid, THEN it is hot"**

A better way to write sentences:

- Hot: **"It is hot"**
- Humid: **"It is humid"**
- Raining: **"It is raining"**

$(o \wedge h) \rightarrow r$ $\quad$ $\text{hot} \wedge \text{humid} \rightarrow \text{raining}$

- The meaning or **semantics** of a sentence determines its **interpretation**
- Given the truth values of all symbols in a sentence, it can be **evaluated** to determine its **truth value** (true or false)
- A **knowledge base** is a list of sentences assumed to be **true**
- A **model** for a KB is a **possible world** (assignment of truth values to propositional symbols) in which each sentence in the KB is **true**
- A sentence p is **satisfiable** if it has some model:
  - $a \land b$ is satisfiable
  - $a \land \neg a$ is **not** satisfiable

- A **tautology** is a sentence that is **true** under all interpretations, no matter what the world is like or how the semantics are defined. Example: “It’s raining or it’s not raining.”

- A **contradiction** is a sentence that is **false** under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”

p	$p \wedge \neg p$	$p \vee \neg p$
F	F	T
T	F	T

More terms

p	q	$\neg p$	$p \land q$	$p \lor q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

- $\neg p$ is a **negation** of p:$\neg \text{am\_wet}$
- $p \land q$ is a **conjunction** of p **and** q: $\text{am\_wet} \land \neg \text{is\_raining}$
- $p \lor q$ is a **disjunction** of p **or** q: $\text{am\_wet} \lor \neg \text{is\_raining}$
- $p \rightarrow q$ is an **implication** of p (**premise**) **implies** q (**conclusion**): $\text{am\_wet} \rightarrow \text{is\_raining}$
- $p \leftrightarrow q$ is a **biconditional** of p **if-and-only-if (iff)** q: $\text{am\_wet} \leftrightarrow \text{is\_raining}$

Simple examples of knowledge base and models:

$\text{is\_raining} \rightarrow \text{am\_wet}$

model: {
$\text{is\_raining} = \boldsymbol{true},$
$\text{am\_wet} = \boldsymbol{true}\}$

$\text{is\_raining} \rightarrow \text{am\_wet}$

$\text{is\_raining} \rightarrow \text{take\_umbrella}$

model 1: {
$\text{is\_raining} = \boldsymbol{false},$
$\text{am\_wet} = \boldsymbol{true},$
$\text{take\_umbrella} = \boldsymbol{true}\}$

model 2: {
$\text{is\_raining} = \boldsymbol{true},$
$\text{am\_wet} = \boldsymbol{true},$
$\text{take\_umbrella} = \boldsymbol{true}\}$

$\text{is\_raining} \rightarrow \text{am\_wet}$

$\text{is\_raining} \land \text{have\_umbrella} \rightarrow \text{open\_umbrella}$

model: {
$\text{is\_raining} = \boldsymbol{true},$
$\text{am\_wet} = \boldsymbol{true},$
$\text{have\_umbrella} = \boldsymbol{true},$
$\text{open\_umbrella} = \boldsymbol{true}\}$

FORM	EQUIVALENCE	NAME
$p \wedge \mathbf{T},\ p \vee \mathbf{F}$	p	**Identity laws**
$p \vee \mathbf{T}$ $p \wedge \mathbf{F}$	$\mathbf{T}$ $\mathbf{F}$	**Domination laws**
$p \vee p,\ \ p \wedge p$	p	**Idempotent laws**
$\neg(\neg p)$	p	**Double negation law**
$p \vee q$ $p \wedge q$	$q \vee p$ $q \wedge p$	**Commutative laws**
$(p \vee q) \vee r$ $(p \wedge q) \wedge r$	$p \vee (q \vee r)$ $p \wedge (q \wedge r)$	**Associative laws**
$p \vee (q \wedge r)$ $p \wedge (q \vee r)$	$(p \vee q) \wedge (p \vee r)$ $(p \wedge q) \vee (p \wedge r)$	**Distributive laws**
$\neg(p \wedge q)$ $\neg(p \vee q)$	$\neg p \vee \neg q$ $\neg p \wedge \neg q$	**De Morgan's laws**
$p \rightarrow q$	$\neg q \rightarrow \neg p$	**Contrapositive equivalence**
$p \vee \neg p$	$\mathbf{T}$	**Excluded middle**
$p \wedge \neg p$	$\mathbf{F}$	**Negation creates opposite**

Let's decide that:

- is_raining means **It's raining outside**
- have_umbrella means **I have an umbrella**
- am_wet means **I am wet**

We can condense propositions replacing them with symbols:

- r means: **It's raining outside**
- u means: **I have an umbrella**
- w means: **I am wet**

EVALUATION

Model: $\{w = \mathbf{T}, r = \mathbf{F}, u = \mathbf{T}\}$

Sentence: $(\neg w \wedge r) \wedge (\neg r \vee u)$

Result of the evaluation: **F**

SATISFACTION

Sentence: $(\neg w \wedge r) \wedge (\neg r \vee u)$

Model: $\{w = ?, r = ?, u = ?\}$

Result of satisfiability:

w	r	u	S
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	F
T	T	F	F
T	T	T	F

- KB **P entails Q**, written $p \vDash q$, means that whenever p is **true**, so is q. In other words, all models of p are also models of q.

  q is entailed by p (a set of premises or assumptions) iff there is no logically possible world in which q is **false** while all the premises in p are **true**.

Example:

- RR1 found the victim ⊨ RR1 looked for the victim
- BR2 was turned off ⊨ BR2 is a robot

Entailment is **stronger** than implication because the former imposes a truth value to the latter. Implication does not do that.

If P is a sentence and K is a knowledge base, a fair question to ask is:
does K entail P?

Or, if more than one knowledge base is available:

$\boldsymbol{KB1}$	$\boldsymbol{KB2}$	$\boldsymbol{KB3}$
$\neg q$	$p \vee q$	$\neg p$
$p \vee q$		$\neg q$

Proof by refutation is a complete inference procedure that is used to prove entailment. It tries to prove something demonstrating the opposite:

- It uses a single inference rule or **resolution**
- The knowledge base is represented in a **Conjunctive Normal Form** (or **CNF**)
- It **reduces** inference to the problem of checking satisfiability

A sentence is in **conjunctive normal form** (or CNF) if it is a **conjunction of disjunction terms**:

- Examples: $\boxed{q} \wedge \boxed{(r \vee s)}$
  $\boxed{p}$
  $\boxed{(p \vee \neg q)} \wedge \boxed{(r \vee s)}$
  $\boxed{(p \vee \neg q)}$
  $\boxed{(p \vee q \vee r)}$

- There is a way to convert a sentence into a clausal form:

SENTENCE	CLAUSAL FORM
$p \leftrightarrow q$	$p \rightarrow q$ $q \rightarrow p$
$p \rightarrow q$	$\neg p \vee q$
$\neg\neg p$	p
$\neg(p \vee q)$	$\neg p \wedge \neg q$
$\neg(p \wedge q)$	$\neg p \vee \neg q$

- The process is performed **conjoining** the KB with the negation of the query
- It's possible to say that KB entails the query iff **the truth table returns false in every row**
- Example:
  Given the following KB:
  1. $w \vee \neg u$
  2. $\neg u \wedge r \rightarrow w$

  S: $r \wedge \neg w$

  Does **KB entail S** (or **$KB \vDash S$**) ?

$$(w \vee \neg u) \wedge (\neg u \wedge r \rightarrow w) \wedge \neg(r \wedge \neg w)$$

$$(w \vee \neg u) \wedge ((\neg u \wedge r) \rightarrow w) \wedge (\neg r \vee \neg\neg w)$$

$$(w \vee \neg u) \wedge (\neg(\neg u \wedge r) \vee w) \wedge (\neg r \vee w)$$

$$(w \vee \neg u) \wedge (u \vee \neg r \vee w) \wedge (\neg r \vee w)$$

$$(w \vee \neg u) \wedge (u \vee \neg r \vee w) \wedge (\neg r \vee w)$$

r	u	w	S
F	F	F	T
F	F	T	T
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

PRACTICE

Exercises from the textbook (chapter 7):

7.1, 7.4, 7.5, 7.7, 7.10

READINGS

Chapters 7

QUESTIONS ?

ARTIFICIAL INTELLIGENCE
COMP 131

FABRIZIO SANTINI

VERSION 3.2