

“Consequently he who wishes to attain to human perfection, must therefore first study Logic, next the various branches of Mathematics in their proper order, then Physics, and lastly Metaphysics.”

Maimonides 1135–1204

- A proposition is a sentence, written in a language, that has a truth value – it is true or false – in a world. A proposition is built from atomic propositions (atoms) and logical connectives.
- Propositions can be built from simpler propositions using logical connectives.

# Propositional Calculus Syntax

- An **atomic proposition** – **atom** – is a symbol, written as sequences of letters, digits, and the underscore (`_`) and start with a lower-case letter.

E.g., `a`, `ai_is_fun`, `lit_1`, `live_outside`, `mimsy`, `sunny`.

- A **proposition** or **logical formula** is either
  - ▶ an atomic proposition or
  - ▶ a **compound proposition** of the form

$\neg p$	“not $p$ ”	<b>negation</b> of $p$
$p \wedge q$	“ $p$ and $q$ ”	<b>conjunction</b> of $p$ and $q$
$p \vee q$	“ $p$ or $q$ ”	<b>disjunction</b> of $p$ and $q$
$p \rightarrow q$	“ $p$ implies $q$ ”	<b>implication</b> of $q$ from $p$
$p \leftarrow q$	“ $p$ if $q$ ”	<b>implication</b> of $p$ from $q$
$p \leftrightarrow q$	“ $p$ if and only if $q$ ”	<b>equivalence</b> of $p$ and $q$
$p \oplus q$	“ $p$ XOR $q$ ”	<b>exclusive-or</b> of $p$ and $q$

where  $p$  and  $q$  are propositions.

- The operators  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftarrow$ ,  $\leftrightarrow$ , and  $\oplus$  are **logical connectives**.

# Why propositions?

- Logical formulae are modular statements of what is (known to be) true
- It is easier to check correctness and debug formulae than tables of what could be true
- We can exploit the Boolean nature for efficient reasoning
- We need a language for asking queries (of what follows in all models) that may be more complicated than asking for the value of a variable
- It is easy to incrementally add formulae
- It can be extended to infinitely many propositions with infinite domains (using logical quantification)

# Semantics of the Propositional Calculus

- An **interpretation** – or **possible world** – is an assignment of true or false to each variable.
- An interpretation is defined by function  $\pi$  that maps **atoms** to  $\{true, false\}$ .  
If  $\pi(a)=true$ , atom  $a$  is **true** in the interpretation.  
If  $\pi(a)=false$ , atom  $a$  is **false** in the interpretation.
- Truth of a compound proposition in an interpretation is defined in terms of the truth of its components:

$p$	$q$	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftarrow q$	$p \leftrightarrow q$	$p \oplus q$
<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>
<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>

- Propositions can have different truth values in different interpretations.

# Models and Logical Consequence

- A **model** of a set of clauses is an interpretation in which all the clauses are *true*.
- If  $KB$  is a set of propositions, proposition  $g$  is a **logical consequence** of  $KB$ , written  $KB \models g$ , if  $g$  is *true* in every model of  $KB$ .
- That is,  $KB \models g$  if there is no interpretation in which  $KB$  is *true* and  $g$  is *false*.

# Simple Example

$$KB = \begin{cases} \text{apple\_eaten} \leftarrow \text{bird\_eats\_apple}. \\ \text{light\_on} \leftarrow \text{night}. \\ \text{night}. \end{cases}$$

	<i>apple_eaten</i>	<i>bird_eats_apple</i>	<i>light_on</i>	<i>night</i>	model of <i>KB</i> ?
<i>l</i> <sub>1</sub>	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>	yes
<i>l</i> <sub>2</sub>	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>	no
<i>l</i> <sub>3</sub>	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>	no
<i>l</i> <sub>4</sub>	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	yes
<i>l</i> <sub>5</sub>	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>	yes

Which of *apple\_eaten*, *bird\_eats\_apple*, *light\_on*, *night* logically follow from *KB*?

$KB \models \text{light\_on}$ ,  $KB \models \text{night}$ ,

$KB \not\models \text{apple\_eaten}$ ,  $KB \not\models \text{bird\_eats\_apple}$

# Human's view of semantics

**Step 1** Begin with a task domain.

**Step 2** Choose atoms in the computer to denote propositions. These atoms have meaning to the KB designer.

**Step 3** Tell the system knowledge about the domain.

**Step 4** Ask the system questions.

— The system can tell you whether the question is a logical consequence.

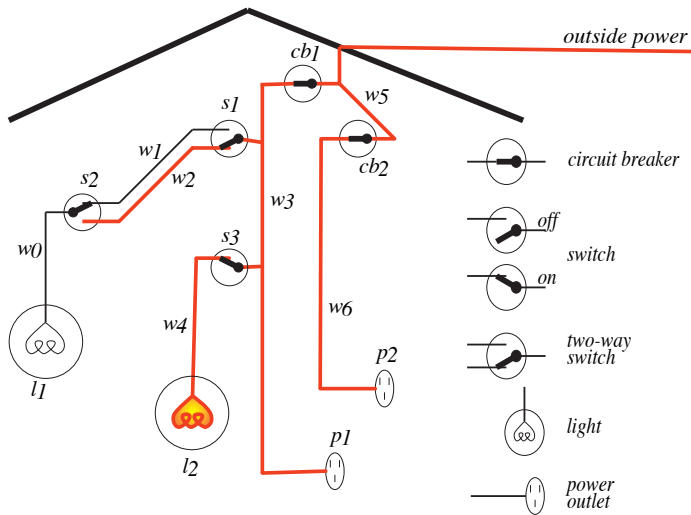
— You can interpret the answer with the meaning associated with the atoms.



# Computer's view of semantics

- The computer doesn't have access to the intended interpretation.
- All it knows is the knowledge base.
- The computer can determine if a formula is a logical consequence of KB.
- If  $KB \models g$  then  $g$  must be true in the intended interpretation.
- If  $KB \not\models g$  then there is a model of  $KB$  in which  $g$  is false. This could be the intended interpretation.

# Electrical Environment



## In computer:

$light2\_broken \leftarrow power\_in\_w\_3$   
 $\wedge sw\_3\_up \wedge unlit\_light2.$

$sw\_3\_up.$

$power\_in\_w\_3 \leftarrow power\_in\_p\_1.$

$unlit\_light2.$

$power\_in\_p\_1.$

## In user's mind:

- $light2\_broken$ : light #2 is broken
- $sw\_3\_up$ : switch 3 is up
- $power\_in\_w\_3$ : there is power in wire 3
- $unlit\_light2$ : light #2 isn't lit
- $power\_in\_p\_1$ : outlet  $p\_1$  has power

---

## Conclusion: $light2\_broken$

- The computer doesn't know the meaning of the symbols
- The user can interpret symbols using their meaning