

COMP3411/9814: Artificial Intelligence

Propositions and Inference

Lecture Outline

- Knowledge Representation and Logic
- Logical Arguments
- Propositional Logic
 - Syntax
 - Semantics
- Validity, Equivalence, Satisfiability, Entailment

Knowledge Bases

- A knowledge base is a set of sentences in a formal language.
- Declarative approach to building an agent:
 - Tell the system what it needs to know,
then it can ask itself what it needs to do
 - Answers should follow from the knowledge based.
- How do you formally specify how to answer questions?

Knowledge Based Agent

The agent must be able to:

- represent states, actions, etc.
- incorporate new percepts
- update internal representations of the world
- deduce hidden properties of the world
- determine appropriate actions

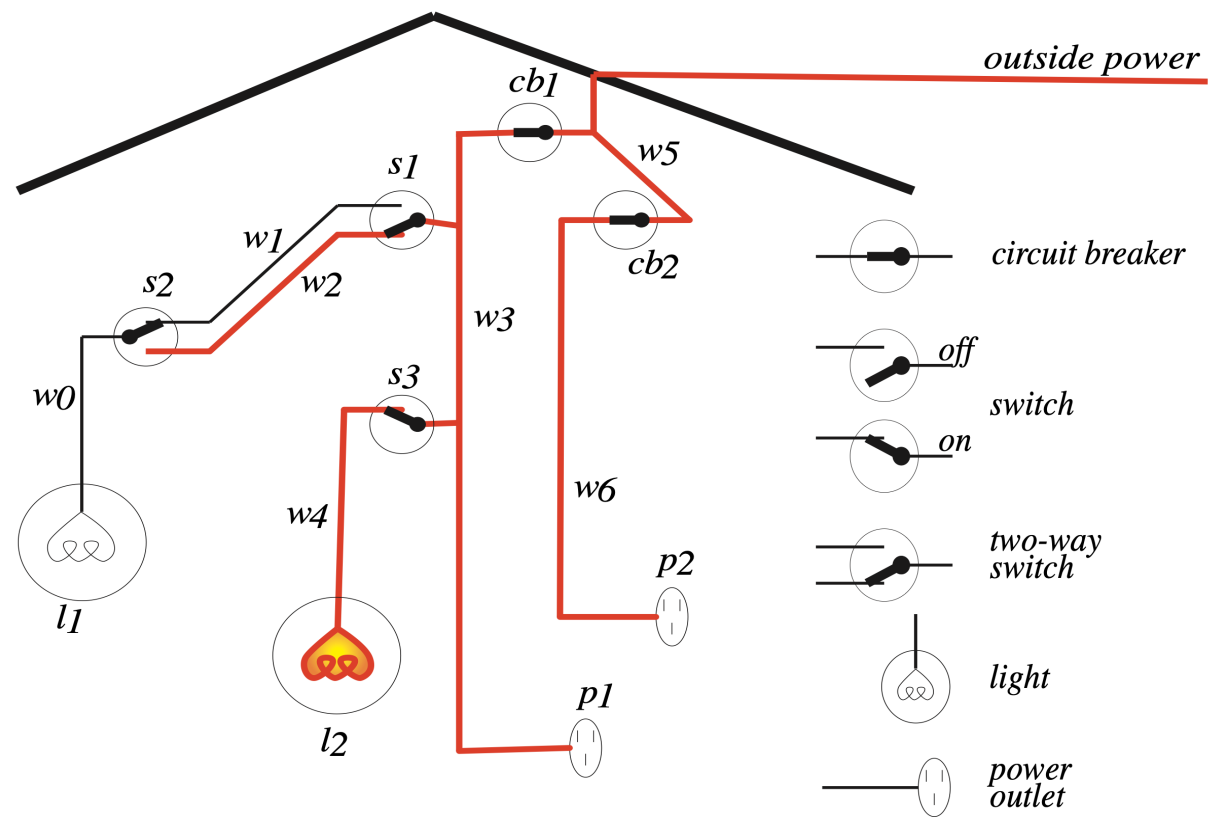
Why Formal Languages (not English, or other natural language)?

- Natural languages are **ambiguous**: “The fisherman went to the bank” (lexical)
- “The boy saw a girl with a telescope” (structural)
- “The table won’t fit through the doorway because it is too [wide/narrow]” (co-reference)
- Ambiguity makes it difficult to interpret meaning of phrases/sentences
 - But also makes inference harder to define and compute
- Symbolic logic is a syntactically unambiguous language

Some Applications of Logic

- In AI
 - Planning and problem solving
 - Natural Language Analysis
- Non-AI
 - Verification of large-scale VLSI
 - Software verification (e.g. SeL4)

Example - Modelling Electrical Circuits



Electrical Circuit in Proposition Logic

$light_l_1.$

$light_l_2.$

$down_s_1.$

$up_s_2.$

$up_s_3.$

$ok_l_1.$

$ok_l_2.$

$ok_cb_1.$

$ok_cb_2.$

$live_outside.$

$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$live_w_0 \leftarrow live_w_1 \wedge up_s_2.$

$live_w_0 \leftarrow live_w_2 \wedge down_s_2.$

$live_w_1 \leftarrow live_w_3 \wedge up_s_1.$

$live_w_2 \leftarrow live_w_3 \wedge down_s_1.$

$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2.$

$live_w_4 \leftarrow live_w_3 \wedge up_s_3.$

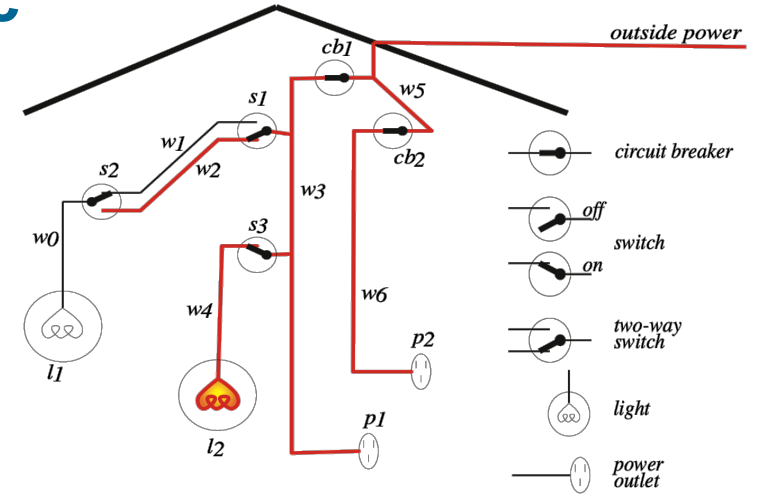
$live_p_1 \leftarrow live_w_3.$

$live_w_3 \leftarrow live_w_5 \wedge ok_cb_1.$

$live_p_2 \leftarrow live_w_6.$

$live_w_6 \leftarrow live_w_5 \wedge ok_cb_2.$

$live_w_5 \leftarrow live_outside.$



$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2$

$lit_l_2 \leftarrow live_w_3 \wedge up_s_3 \wedge ok_l_2$

$lit_l_2 \leftarrow live_w_5 \wedge ok_cb_1 \wedge up_s_3 \wedge ok_l_2$

$lit_l_2 \leftarrow live_outside \wedge ok_cb_1 \wedge up_s_3 \wedge ok_l_2$

$lit_l_2 \leftarrow ok_cb_1 \wedge up_s_3 \wedge ok_l_2$

$lit_l_2 \leftarrow up_s_3 \wedge ok_l_2$

$lit_l_2 \leftarrow ok_l_2$

lit_l_2

Electrical Circuit in Proposition Logic

$light_l_1.$

$light_l_2.$

$down_s_1.$

$up_s_2.$

$up_s_3.$

$ok_l_1.$

$ok_l_2.$

$ok_cb_1.$

$ok_cb_2.$

$live_outside.$

$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$live_w_0 \leftarrow live_w_1 \wedge up_s_2.$

$live_w_0 \leftarrow live_w_2 \wedge down_s_2.$

$live_w_1 \leftarrow live_w_3 \wedge up_s_1.$

$live_w_2 \leftarrow live_w_3 \wedge down_s_1.$

$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2.$

$live_w_4 \leftarrow live_w_3 \wedge up_s_3.$

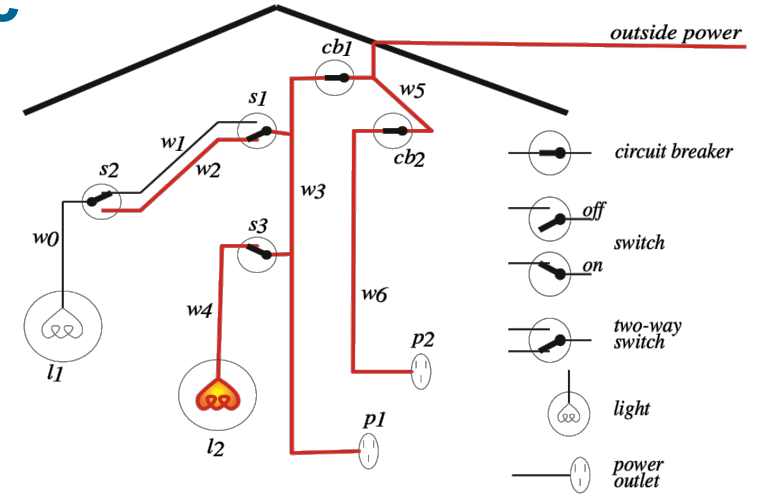
$live_p_1 \leftarrow live_w_3.$

$live_w_3 \leftarrow live_w_5 \wedge ok_cb_1.$

$live_p_2 \leftarrow live_w_6.$

$live_w_6 \leftarrow live_w_5 \wedge ok_cb_2.$

$live_w_5 \leftarrow live_outside.$



$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_3 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_5 \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_outside \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow up_s_1 \wedge up_s_2 \wedge ok_l_1$

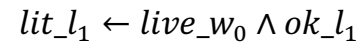
$fail$

$light_{l_1}.$
$$down_{s_1}.$$

*up*_{s3}.

 $ok_{l_2}.$

ok_cb2.

$$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$$
$$live_w_0 \leftarrow live_w_1 \wedge up_s_2.$$
$$live_w_0 \leftarrow live_w_2 \wedge down_s_2.$$
$$live_w_1 \leftarrow live_w_3 \wedge up_s_1.$$
$$live_w_2 \leftarrow live_w_3 \wedge down_s_1.$$
$$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2.$$
$$live_w_4 \leftarrow live_w_3 \wedge up_s_3.$$
$$live_p_1 \leftarrow live_w_3.$$
$$live_w_3 \leftarrow live_w_5 \wedge ok_cb_1.$$
$$live_p_2 \leftarrow live_w_6.$$
$$live_w_6 \leftarrow live_w_5 \wedge ok_cb_2.$$
$$live_w_5 \leftarrow live_outside.$$

$$lit_l_1 \leftarrow live_w_1 \wedge up_s_2 \wedge ok_l_1$$
$$lit_l_1 \leftarrow live_w_3 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$$
$$lit_l_1 \leftarrow live_w_5 \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$$
$$lit_l_1 \leftarrow live_outside \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$$
$$lit_l_1 \leftarrow ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$$
$$lit_l_1 \leftarrow up_s_1 \wedge up_s_2 \wedge ok_l_1$$

fail

Electrical Circuit in Proposition Logic

$light_l_1.$

$light_l_2.$

$down_s_1.$

$up_s_2.$

$up_s_3.$

$ok_l_1.$

$ok_l_2.$

$ok_cb_1.$

$ok_cb_2.$

$live_outside.$

$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$live_w_0 \leftarrow live_w_1 \wedge up_s_2.$

$live_w_0 \leftarrow live_w_2 \wedge down_s_2.$

$live_w_1 \leftarrow live_w_3 \wedge up_s_1.$

$live_w_2 \leftarrow live_w_3 \wedge down_s_1.$

$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2.$

$live_w_4 \leftarrow live_w_3 \wedge up_s_3.$

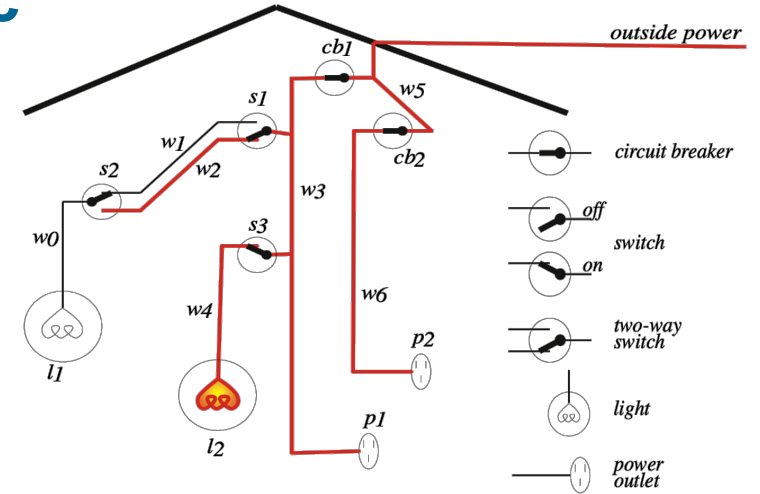
$live_p_1 \leftarrow live_w_3.$

$live_w_3 \leftarrow live_w_5 \wedge ok_cb_1.$

$live_p_2 \leftarrow live_w_6.$

$live_w_6 \leftarrow live_w_5 \wedge ok_cb_2.$

$live_w_5 \leftarrow live_outside.$



$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_2 \wedge down_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_3 \wedge down_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_5 \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_2 \leftarrow live_outside \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow up_s_1 \wedge up_s_2 \wedge ok_l_1$

$fail$

Electrical Circuit in Proposition Logic

$light_l_1.$

$light_l_2.$

$down_s_1.$

$up_s_2.$

$up_s_3.$

$ok_l_1.$

$ok_l_2.$

$ok_cb_1.$

$ok_cb_2.$

$live_outside.$

$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$live_w_0 \leftarrow live_w_1 \wedge up_s_2.$

$live_w_0 \leftarrow live_w_2 \wedge down_s_2.$

$live_w_1 \leftarrow live_w_3 \wedge up_s_1.$

$live_w_2 \leftarrow live_w_3 \wedge down_s_1.$

$lit_l_2 \leftarrow live_w_4 \wedge ok_l_2.$

$live_w_4 \leftarrow live_w_3 \wedge up_s_3.$

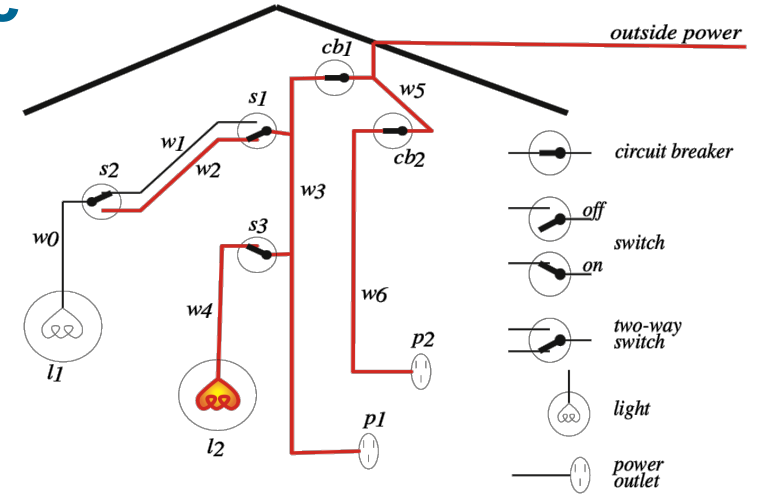
$live_p_1 \leftarrow live_w_3.$

$live_w_3 \leftarrow live_w_5 \wedge ok_cb_1.$

$live_p_2 \leftarrow live_w_6.$

$live_w_6 \leftarrow live_w_5 \wedge ok_cb_2.$

$live_w_5 \leftarrow live_outside.$



$lit_l_1 \leftarrow live_w_0 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_2 \wedge down_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_3 \wedge down_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow live_w_5 \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_2 \leftarrow live_outside \wedge ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow ok_cb_1 \wedge up_s_1 \wedge up_s_2 \wedge ok_l_1$

$lit_l_1 \leftarrow up_s_1 \wedge up_s_2 \wedge ok_l_1$

$fail$

Syntax vs Semantics

Syntax - legal sentences in knowledge representation language
(e.g. in the language of arithmetic expressions $x < 4$)

Semantics - meaning of sentences.

Refers to a sentence's relationship to the “real world” or to some model of the world.

- Semantic properties of sentences include truth and falsity
(e.g. $x < 4$ is true for $x = 3$ and false when $x = 5$).
- Semantic properties of names and descriptions include referents.
- The meaning of a sentence is not intrinsic to that sentence.
 - An interpretation is required to determine sentence meanings.
 - Interpretations are agreed amongst a linguistic community.

Propositions

- Propositions are entities (facts or non-facts) that can be true or false

Examples:

- “The sky is blue” - the sky is blue (here and now).
- “Socrates is bald” (assumes ‘Socrates’, ‘bald’ are well defined)
“The car is red” (requires ‘the car’ to be identified)
- “Socrates is bald and the car is red” (complex proposition)
- Use single letters to represent propositions, e.g. P : Socrates is bald
- Reasoning is independent of definitions of propositions

Logical Arguments

An **argument** relates a set of premises to a conclusion

- **valid** if the conclusion **necessarily follows** from the premises

All humans have 2 eyes

Jane is a human

Therefore Jane has 2 eyes

All humans have 4 eyes

Jane is a human

Therefore Jane has 4 eyes

- Both are (logically) correct valid arguments
- Which statements are true/false? Why?

Logical Arguments

An **argument** relates a set of premises to a conclusion

- **invalid** if the conclusion can be false when the premises are all true

All humans have 2 eyes

Jane has 2 eyes

Therefore Jane is human

No human has 4 eyes

Jane has 2 eyes

Therefore Jane is not human

- Both are (logically) **incorrect invalid** arguments
- Which statements are true/false? Why?

Propositional Logic

- Letters stand for “basic” propositions
- Combine into more complex sentences using operators **not**, **and**, **or**, **implies**, **iff**
- Propositional **connectives**:

\neg	negation	$\neg P$	“not P”
\wedge	conjunction	$P \wedge Q$	“P and Q”
\vee	disjunction	$P \vee Q$	“P or Q”
\rightarrow	implication	$P \rightarrow Q$	“If P then Q”
\leftrightarrow	bi-implication	$P \leftrightarrow Q$	“P if and only if Q”

From English to Propositional Logic

- “It is not the case that the sky is blue”: $\neg B$
(alternatively “the sky is not blue”)
- “The sky is blue and the grass is green”: $B \wedge G$
- “Either the sky is blue or the grass is green”: $B \vee G$
- “If the sky is blue, then the grass is not green”: $B \rightarrow \neg G$
- “The sky is blue if and only if the grass is green”: $B \leftrightarrow G$
- “If the sky is blue, then if the grass is not green, the plants will not grow”:
 $B \rightarrow (\neg G \rightarrow \neg P)$

Operator Precedence

- $(P \rightarrow (Q \rightarrow (\neg(R))))$ vs $P \rightarrow (Q \rightarrow \neg R)$
- Rules for omitting parentheses
 - Omit parentheses where possible
 - Precedence from highest to lowest is: \neg , \wedge , \vee , \rightarrow , \leftrightarrow
 - All binary operators are left associative
 - so $P \rightarrow Q \rightarrow R$ abbreviates $(P \rightarrow Q) \rightarrow R$
- Sometimes parentheses can't be removed:
 - Is $(P \vee Q) \vee R$ (always) the same as $P \vee (Q \vee R)$?
 - Is $(P \rightarrow Q) \rightarrow R$ (always) the same as $P \rightarrow (Q \rightarrow R)$? **NO!**
- <https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

P	Q	R	$((P \rightarrow Q) \rightarrow R)$	$(P \rightarrow (Q \rightarrow R))$
F	F	F	F	T
F	F	T	T	T
F	T	F	F	T
F	T	T	T	T
T	F	F	T	T
T	F	T	T	T
T	T	F	F	F
T	T	T	T	T

Truth Table Semantics

- The semantics of the connectives can be given by **truth tables**

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	True	True	False	True	True	False
False	False	True	False	False	True	True

- One row for each possible assignment of True/False to variables
- Important: P and Q are any sentences, including complex sentences

Example – Complex Sentence

R	S	$\neg R$	$R \wedge S$	$\neg R \vee S$	$(R \wedge S) \rightarrow (\neg R \vee S)$
True	True	False	True	True	True
True	False	False	False	False	True
False	True	True	False	True	True
False	False	True	False	True	True

Thus $(R \wedge S) \rightarrow (\neg R \vee S)$ is a **tautology**

<https://web.stanford.edu/class/cs103/tools/truth-table-tool/>

Definitions

- A sentence is **valid** if it is True under all possible assignments of True/False to its variables (e.g. $P \vee \neg P$)
- A **tautology** is a valid sentence
- Two sentences are **equivalent** if they have the same truth table, e.g. $P \wedge Q$ and $Q \wedge P$
 - ▶ So P is equivalent to Q if and only if $P \leftrightarrow Q$ is valid
- A sentence is **satisfiable** if there is **some** assignment of True/False to its variables for which the sentence is True
- A sentence is **unsatisfiable** if it is not satisfiable (e.g. $P \wedge \neg P$)
 - ▶ Sentence is False for all assignments of True/False to its variables

Material Implication

- $P \rightarrow Q$ evaluates to False only when P is True and Q is False
- $P \rightarrow Q$ is equivalent to $\neg P \vee Q$: material implication
- English usage often suggests a causal connection between antecedent (P) and consequent (Q) – this is not reflected in the truth table
- All these are tautologies
 - ▶ $(P \wedge Q) \rightarrow Q$
 - ▶ $P \rightarrow (P \vee Q)$
 - ▶ $(P \wedge \neg P) \rightarrow Q$

Material Implication

- $P \rightarrow Q$ evaluates to False only when P is True and Q is False
- $P \rightarrow Q$ is equivalent to $\neg P \vee Q$: material implication
- English usage often suggests a causal connection between antecedent (P) and consequent (Q) – this is not reflected in the truth table
- All these are tautologies
 - ▶ $(P \wedge Q) \rightarrow Q = \neg(P \wedge Q) \vee Q = \neg P \vee \neg Q \vee Q = T$
 - ▶ $P \rightarrow (P \vee Q) = \neg P \vee P \vee Q = T$
 - ▶ $(P \wedge \neg P) \rightarrow Q = \neg(P \wedge \neg P) \vee Q = \neg P \vee P \vee Q = T$

Logical Equivalences – All Valid

Commutativity: $p \wedge q \leftrightarrow q \wedge p$

$$p \vee q \leftrightarrow q \vee p$$

Associativity: $p \wedge (q \wedge r) \leftrightarrow (p \wedge q) \wedge r$

$$p \vee (q \vee r) \leftrightarrow (p \vee q) \vee r$$

Distributivity: $p \wedge (q \vee r) \leftrightarrow (p \wedge q) \vee (p \wedge r)$

$$p \vee (q \wedge r) \leftrightarrow (p \vee q) \wedge (p \vee r)$$

Implication: $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$

Idempotent: $p \wedge p \leftrightarrow p$

$$p \vee p \leftrightarrow p$$

Double negation: $\neg \neg p \leftrightarrow p$

Contradiction: $p \wedge \neg p \leftrightarrow \text{FALSE}$

Excluded middle:

$$p \vee \neg p \leftrightarrow \text{TRUE}$$

De Morgan: $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$

$$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$$

Proof of Equivalence

Let $P \Leftrightarrow Q$ mean “ P is equivalent to Q ” ($P \Leftrightarrow Q$ is not a formula)

Then $P \wedge (Q \rightarrow R) \Leftrightarrow \neg(P \rightarrow Q) \vee (P \wedge R)$

$P \wedge (Q \rightarrow R) \Leftrightarrow P \wedge (\neg Q \vee R)$	[Implication]
$\Leftrightarrow (P \wedge \neg Q) \vee (P \wedge R)$	[Distributivity]
$\Leftrightarrow (\neg \neg P \wedge \neg Q) \vee (P \wedge R)$	[Double negation]
$\Leftrightarrow \neg(\neg P \vee Q) \vee (P \wedge R)$	[De Morgan]
$\Leftrightarrow \neg(P \rightarrow Q) \vee (P \wedge R)$	[Implication]

Assumes substitution: if $A \Leftrightarrow B$, replace A by B in any subformula

Assumes equivalence is transitive: if $A \Leftrightarrow B$ and $B \Leftrightarrow C$ then $A \Leftrightarrow C$

Interpretations and Models

- An **interpretation** is an assignment of values to all variables.
- A **model** is an interpretation that satisfies the constraints.
 - A model is a **possible world** in which a sentence (or set of sentences) is true, e.g.
 - $x + y = 4$ in a world where $x = 2$ and $y = 2$
 - May be more than one possible world (e.g. $x = 3$ and $y = 1$)
- Often want to know what is true in all models.
- A proposition is statement that is true or false in each interpretation.

Entailment

- Entailment means that one sentence follows logically from another sentence, or set of sentences (i.e. a knowledge base):

$$KB \models \alpha$$

- Knowledge base KB entails sentence α if and only if α is true in all models (possible worlds) where KB is true.

e.g. the KB containing “the Moon is full” and “the tide is high” entails “Either the Moon is full or the tide is high”.

$$\text{e.g. } x + y = 4 \text{ entails } 4 = x + y$$

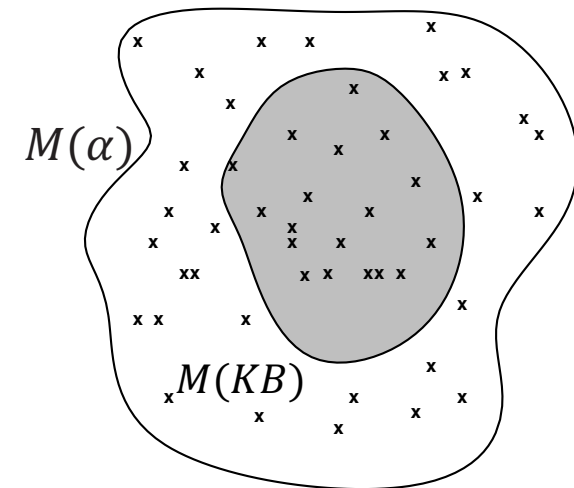
- Entailment is a relationship between sentences based on semantics.

Models

- For propositional logic, a model is **one** row of the truth table
- A model M **is a model of** a sentence α if α is True in M

Let $M(\alpha)$ be the set of all models of α

Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$



Entailment

- S **entails** P ($S \models P$) if whenever all formulae in S are True, P is True
 - ▶ Semantic definition – concerns truth (not proof)
- Compute whether $S \models P$ by calculating a truth table for S and P
 - ▶ Syntactic notion – concerns computation/proof
 - ▶ Not always this easy to compute (how inefficient is this?)
- A tautology is a special case of entailment where S is the empty set
 - ▶ All rows of the truth table are True

Entailment Example

P	Q	$P \rightarrow Q$	Q
True	True	True	True
True	False	False	False
False	True	True	True
False	False	True	False

- $\{P, P \rightarrow Q\} \models Q$ since when both P and $P \rightarrow Q$ are True (row 1), Q is also True
- $P \rightarrow Q$ is calculated from P and Q using the truth table definition, and Q is used again to check the entailment

Example – $S \models P$

Each row is an interpretation of S .
Only the first row is a model of S .

$$S = \{p \rightarrow q, q \rightarrow p, p \vee q\}$$

$$P = p \wedge q$$

p	q	$p \rightarrow q$	$q \rightarrow p$	$p \vee q$	S	$p \wedge q$
T	T	T	T	T	T	T
T	F	F	T	T	F	F
F	T	T	F	T	F	F
F	F	T	T	F	F	F

Example – $S \models P$

$$S = \{q \vee r, q \rightarrow \neg p, \neg(r \wedge p)\}$$

$$P = \neg p$$

p	q	r	$q \vee r$	$q \rightarrow \neg p$	$\neg(r \wedge p)$	S	$\neg p$
T	T	T	T	F		F	
T	T	F	T	F		F	
T	F	T	T	T	F	F	
T	F	F	F			F	
F	T	T	T	T	T	T	T
F	T	F	T	T	T	T	T
F	F	T	T	T	T	T	T
F	F	F	F			F	

Conclusion

- Ambiguity of natural languages avoided with formal languages
- Enables formalisation of (truth preserving) entailment
- Propositional Logic: Simplest logic of truth and falsity
- Knowledge Based Systems: First-Order Logic
- Automated Reasoning: How to compute entailment (inference)
- Many logics not studied in this course