Techniques in Artificial Intelligence

Propositional Logic

Iyad Rahwan

irahwan@masdar.ac.ae

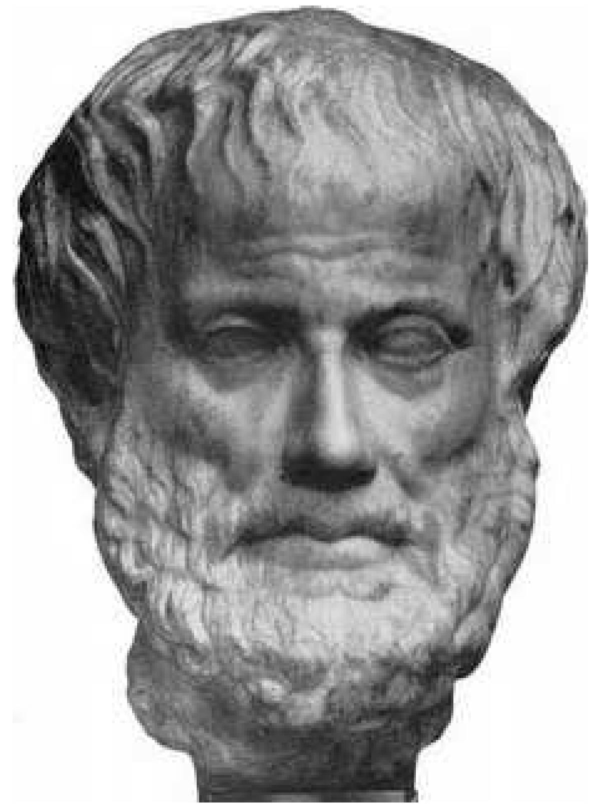
Masdar Institute of Science & Technology

This Lecture

- Logic as a Representation Language
- Soundness and Completeness of Inference
- Propositional Logic
  - Syntax
  - Semantics & Truth Tables
  - Natural Deduction
  - Inference with truth-table enumeration

Logic as a Representation Language

- **Logic:** A branch of philosophy concerned with the study of correct inference (or argument)
- Dates back at least to Aristotle
- **Symbolic logic:** is the study of symbolic abstractions that capture the formal features of logical inference.

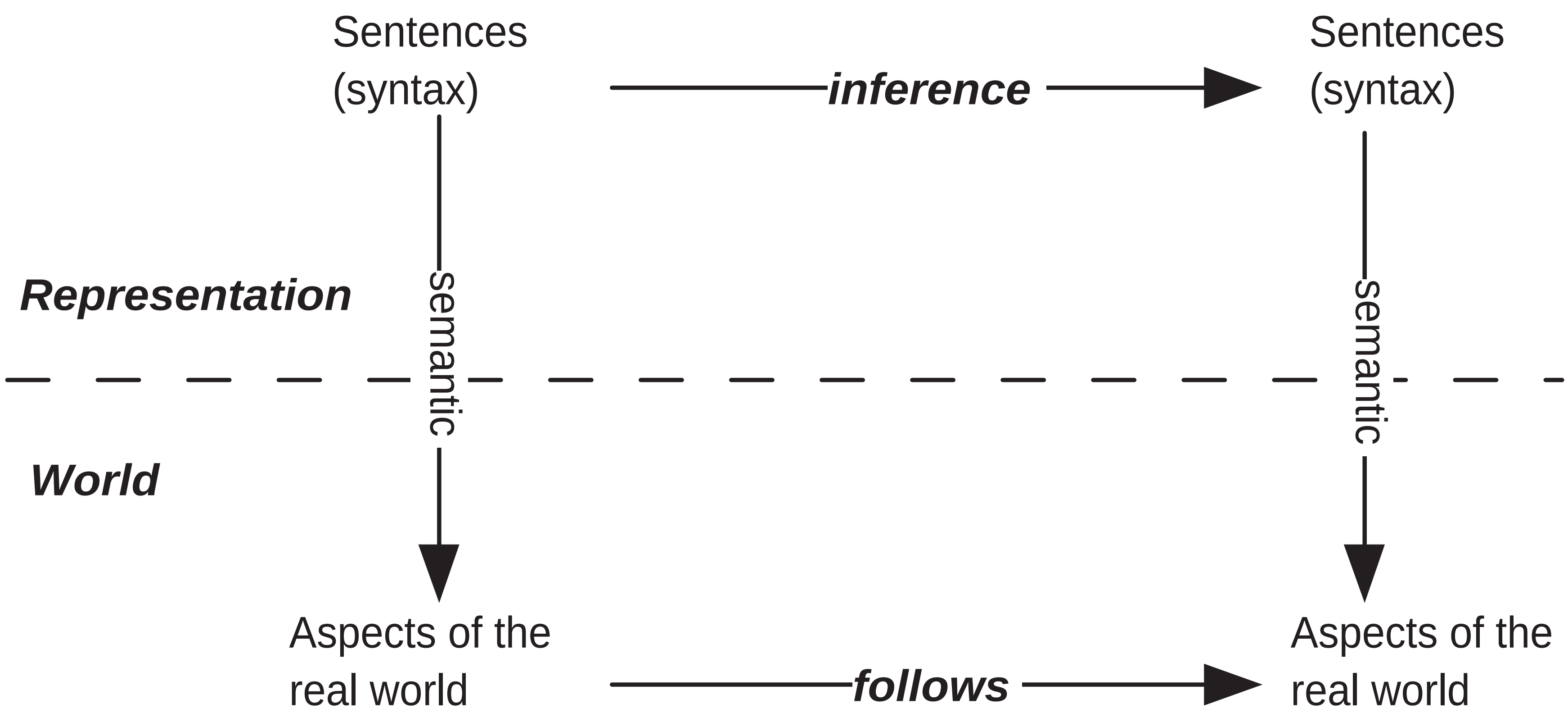


Aristotle (384 BC - 322 BC)

Logic as a Representation Language

- **Syntax:** possible configurations (e.g. a C++ or Prolog program) that can constitute sentences.
- **Reasoning (or inference or proof theory):** a process of deriving new configurations (e.g. conclusions) from old ones (e.g. evidence). It works on the *syntactic* level.
  - Expression "$S_1 \vdash S_2$" means we derived sentences S_2 from sentences S_1 using inference procedure $\vdash$
- **Semantics:** The facts in the world to which the sentences refer.
  - Expression "$S_1 \models S_2$" means aspects of the world S_2 *actually* follow from aspects S_1
  - We say that S_1 **entails** (or **semantically entails**) S_2

Syntax and Semantics



Interpretation and Semantics

- An **interpretation** of a set of sentences is a *specific state* of the world that the sentences refer to.
- Sometimes, a set of sentences may have more than one interpretation.
- In this case, the semantics of a set of sentences can be defined as a *set of possible interpretations* which are consistent with the sentences
- Any world in which a sentence is true under a particular interpretation is called a **model** of that sentence under that interpretation.

Validity and Satisfiability

- A sentence is **satisfiable** if it is true under some interpretation.
  - E.g. “It is raining”
- A sentence is **unsatisfiable** (or **contradictory**) if it is not true under any interpretation.
  - E.g. “It is raining and it is not raining”
- A sentence is **valid** if it is true all possible interpretations.
  - E.g. “It’s either raining or not raining”
- A set of sentences is **consistent** if they *can* be true together; i.e. if there is some interpretation under which they are *together* satisfiable.
  - E.g. “It’s raining and I am hungry”

Properties of Inference Procedures

- **Soundness:** An inference procedure $\vdash$ is *sound* if it generates only entailed sentences.
  - i.e. every sentence s derived using the inference procedure from a set of sentences S_1 is also entailed by S_1 in the real world
  - i.e. whenever $S_1 \vdash s$, we also have $S_1 \models s$
- **Completeness:** An inference procedure $\vdash$ is *complete* if it can generate all entailed sentences.
  - i.e. every sentence s that is entailed by S_1 in the real world can also be derived from S_1 using the inference procedure
  - i.e. whenever $S_1 \models s$, we also have $S_1 \vdash s$

Types of Logic

- **Monotonic Logic:** A logic in which adding a sentence to the set of given sentences never invalidates a previously derived sentence
  - i.e. if $S \vdash s$, then $(S \cup S') \vdash s$ regardless of the content of S'
- **Non-monotonic Logic:**
  - if $S \vdash s$, then it may be the case that $(S \cup S') \vdash (not)s$ for some new sentences S'
- Non-monotonic logics are more natural, but harder to automate

Syntax of Propositional Logic

$$
\begin{array}{rcl}
Sentence & ::= & AtomicSentence \\
 & & \mid CompoundSentence \\
AtomicSentence & ::= & \textbf{True} \mid \textbf{False} \mid P \mid Q \mid R \mid \ldots \\
CompoundSentence & ::= & Sentence\ Connective\ Sentence \\
 & & \mid \neg Sentence \mid (Sentence) \\
Connective & ::= & \wedge \mid \vee \mid \rightarrow \mid \leftrightarrow
\end{array}
$$

Syntactic ambiguities are resolved using () and the precendence order $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Sometimes we refer to atomic sentences as “propositions”

Semantics of Propositional Logic

- Meaning of atomic sentences P is the interpretation of that sentence to either **True** or **False**
- E.g. if P stands for “BUiD is in Dubai” then the meaning of P is either **True** or **False** depending on whether BUiD is actually in Dubai
- Meaning of compound sentences is given as follows:
  - $\neg P$ is **True** iff P is **False**
  - $P \wedge Q$ is **True** iff P is **True** and Q is **True**
  - $P \vee Q$ is **True** iff P is **True** or Q is **True**
  - $P \rightarrow Q$ is **True** iff P is **False** or Q is **True**
  - $P \leftrightarrow Q$ is **True** iff $P \rightarrow Q$ is **True** and $Q \rightarrow P$ is **True**

Example 1

Let:

- P denote: “I am a BUiD student”
- Q denote: “I study KR”
- R denote: “I am cool”
- S denote: “I listen to Metallica”
- T denote: “I work hard”

$(P \wedge Q) \rightarrow R$ means: “if I am a BUiD student, and I study KR, then I am cool”

$((P \wedge Q) \vee S) \rightarrow R$ means: ?

$((P \wedge Q \wedge \neg T) \vee \neg S) \rightarrow \neg R$ means: ?

Semantics Continued: Interpretation

An *interpretation of propositional logic* is an assignment of either **True** or **False** to each proposition.

For example, we can specify an interpretation I_1:

$$I_1 = \{P = \textbf{True}, Q = \textbf{True}, R = \textbf{True}, S = \textbf{False}, T = \textbf{True}\}$$

Since each proposition can take one of two values (**True** or **False**), then given n propositions, we have 2^n possible interpretations (i.e. 2^n possible truth assignments).

In Example 1, we have 5 propositions, therefore $2^5 = 32$ possible interpretations.

Semantics Continue: Truth Tables

A *Truth Table* is a table of all possible interpretations of the propositions in the language.

P	Q
True	True
True	False
False	True
False	False

Truth Tables (cont.)

We can use truth tables to specify semantics of connectives.

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

This can then be used to specify the semantics of compound sentences.

Truth Tables: Example

Present the truth table for the statement $((P \vee Q) \wedge \neg Q) \to P$

P	Q	$P \vee Q$	$(P \vee Q) \wedge \neg Q$	$((P \vee Q) \wedge \neg Q) \to P$
True	True	True	False	**True**
True	False	True	True	**True**
False	True	True	False	**True**
False	False	False	False	**True**

Validity, Satisfiability & Contradiction

Validity: A sentence is *valid* (i.e. *necessarily true*) if it is true in all interpretations
E.g. $((P \vee Q) \wedge \neg Q) \rightarrow P$ (from the previous slide) is valid.
E.g. $P \vee \neg P$ is valid
Valid sentences are also called *tautologies* or *theorems*

Satisfiability: A sentence is *satisfiable* if there is some interpretation for which the sentence is true

Contradiction: A sentence is *contradictory* (i.e. *impossibly true*) if there is no interpretation in which it is true
E.g. $P \wedge \neg P$ is contradictory

Models

Models: A *model* of a sentence is an interpretation in which the sentences is true.

- E.g. the interpretation $I = \{P = False, Q = True\}$ is a model of the sentence $P \vee Q$

Equivalence: Two sentences are *equivalent* if they are true in the same set of models.

- E.g. $P \wedge Q$ is equivalent to $Q \wedge P$
- We write $(P \wedge Q) \leftrightarrow (Q \wedge P)$

Inference

How do we decide whether a sentence is valid?

- Generate a truth table and see whether the sentence is true in all interpretations

However, the size of the table grows exponentially with the number of propositions!

Are there other ways?

Yes! But we won't be looking at all of them.

One notable one is *natural deduction*

Natural Deduction

In natural deduction, we have *inference rules*: general computable pattern for going from true sentences to true sentences.

Inference rules allow us to *infer* (or *deduce*) sentences (i.e. formulas) from other sentences.

Given a set of *premises* (i.e. assumed sentences), we can infer some other sentences (i.e. conclusions)

The expression $\psi_1, \psi_2, \ldots, \psi_n \vdash \alpha$ denotes that if we assume the truth of sentences $\psi_1, \psi_2, \ldots, \psi_n$ then we can infer α.

I.e. α is true in every model of $\psi_1, \psi_2, \ldots, \psi_n$

Rules of Natural Deduction

$\wedge$ **introduction ($\wedge$i):** $\frac{A \qquad B}{A \wedge B}$
In words, if A and B are true, then $A \wedge B$ is true

$\wedge$ **elimination ($\wedge$e1):** $\frac{A \wedge B}{A}$

$\wedge$ **elimination ($\wedge$e2):** $\frac{A \wedge B}{B}$

$\vee$ **introduction ($\vee$i1):** $\frac{A}{A \vee B}$

$\vee$ **introduction ($\vee$i2):** $\frac{B}{A \vee B}$

Rules of Natural Deduction (cont.)

If we can deduce C from A and we can deduce C from B, then we can say that we can deduce C from $A \vee B$:

$\vee$ **elimination ($\vee$e):**

$$\frac{A \vee B \quad \begin{matrix} A \\ \vdots \\ C \end{matrix} \quad \begin{matrix} B \\ \vdots \\ C \end{matrix}}{C}$$

Rules of Natural Deduction (cont.)

If we can deduce B from A, then we can write $A \rightarrow B$:

$\rightarrow$ introduction ($\rightarrow$i):

$$\dfrac{\begin{array}{c} A \\ . \\ . \\ B \end{array}}{A \rightarrow B}$$

$\rightarrow$ elimination ($\rightarrow$e): $\dfrac{A \qquad A \rightarrow B}{B}$

This is also called *modus ponens*

Rules of Natural Deduction (cont.)

If we can deduce $\perp$ (contradiction) from A, then we have deduced that $\neg A$ (Sometimes called *proof-by-contradiction*):
Reductio Ad Absurdum (RAA):

$$\frac{\begin{array}{c} A \\ \cdot \\ \cdot \\ \perp \end{array}}{\neg A}$$

$\neg$ **elimination ($\neg$e):** $\frac{A \qquad \neg A}{\perp}$

I.e. we deduce a contradiction if something is both true and false

Rules of Natural Deduction (cont.)

If you negate twice, you get the same thing:

$\neg\neg$ **introduction ($\neg\neg$i):**

$$\frac{A}{\neg\neg A}$$

$\neg\neg$ **elimination ($\neg\neg$e):**

$$\frac{\neg\neg A}{A}$$

If you deduce a contradiction, then anything can be deduced (see next slide for why!)

$\perp$ **elimination ($\perp$e):**

$$\frac{\perp}{A}$$

Rules of Natural Deduction (cont.)

Why do we infer anything from $\perp$?

Contradiction means we have P and $\neg P$ for some proposition P

But from P we can infer $P \vee Q$ for any Q we like (by rule $\vee$**i1**)

And from $\neg P$ and $P \vee Q$ we can derive Q

Therefore, once we have a contradiction between a proposition and its negation, we can derive any other proposition (or formula).

Natural Deduction Example

Show that $P, \neg\neg(Q \wedge R) \vdash \neg\neg P \wedge R$

Solution:

1	P	premise
2	$\neg\neg(Q \wedge R)$	premise
3	$\neg\neg P$	$\neg\neg$i
4	$Q \wedge R$	$\neg\neg$e on 2
5	R	$\wedge$e on 4
6	$\neg\neg P \wedge R$	$\wedge$i on 3 and 5

Exercise? Try to show this using a truth table!

Inference using natural deduction rules is sound and complete (and some mathematician has proved this!)

Natural Deduction and Truth Tables

Recall: The expression $\psi_1, \psi_2, \ldots, \psi_n \vdash \alpha$ denotes that if we assume the truth of sentences $\psi_1, \psi_2, \ldots, \psi_n$ then we can infer α.

I.e. α is true in every model of $\psi_1, \psi_2, \ldots, \psi_n$

I.e. Whenever $\psi_1, \psi_2, \ldots, \psi_n$ are true in the truth table, α will also be true (but not necessarily vice versa!).

$\vdash \alpha$ means that we can use natural deduction to show that sentence α is always true (i.e. without assuming anything).

In other words, $\vdash \alpha$ means that sentence α is valid (i.e. a tautology, i.e. a theorem).

Natural Deduction and Truth Tables

How can we show that $\psi_1, \psi_2, \ldots, \psi_n \vdash \alpha$ using a truth table instead of natural deduction?

1. Construct a truth table for sentence $\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$,
2. Extend the truth table to show the values of sentence α.
3. Show that whenever $\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n$ is true, then α is also true.
   - Or, equivalently, show that $(\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n) \rightarrow \alpha$ is valid (i.e. always true)

Example

Show that $P, \neg\neg(Q \wedge R) \vdash \neg\neg P \wedge R$ using a truth table.

P	Q	R	$Q \wedge R$	$P \wedge \neg\neg(Q \wedge R)$	$\neg\neg P \wedge R$
True	True	True	True	**True**	**True**
True	True	False	False	False	False
True	False	True	False	False	*True*
True	False	False	False	False	False
False	True	True	True	False	False
False	True	False	False	False	False
False	False	True	False	False	False
False	False	False	False	False	False

Whenever $P \wedge \neg\neg(Q \wedge R)$ is true, so is $\neg\neg P \wedge R$

I.e. $(P \wedge \neg\neg(Q \wedge R)) \rightarrow \neg\neg P \wedge R$ is valid

Truth-Table Enumeration Algorithm

Algorithm to decide if α is entailed by KB

$symbols \Leftarrow$ *list of proposition symbols in* KB *and* α

$model \Leftarrow []$

function: *TT-CHECK-ALL*$(KB, \alpha, symbols, model)$ returns $true$ or $false$

if *EMPTY*$(symbols)$ **then**

  if *PL-TRUE*$(KB, model)$ **then**

    return *PL-TRUE*$(\alpha, model)$

  else

    return $true$

  end if

else

  $P \Leftarrow$ *FIRST*$(symbols)$; $rest \Leftarrow$ *REST*$(symbols)$

  return *TT-CHECK-ALL*$(KB, \alpha, rest,$ *EXTEND*$(P, true, model))$ and *TT-CHECK-ALL*$(KB, \alpha, rest,$ *EXTEND*$(P, false, model))$

end if

TT Enumeration Algorithm (cont.)

- Recursive enumeration of a finite space of assignments to variables
- *PL-TRUE*$()$ returns true if a sentence holds within a model.
- Variable $model$ represents a partial model: an assignment to only some of the variables
- Function *EXTEND*$(P, x, model)$ returns a new partial model in which P has the value $x \in \{true, false\}$

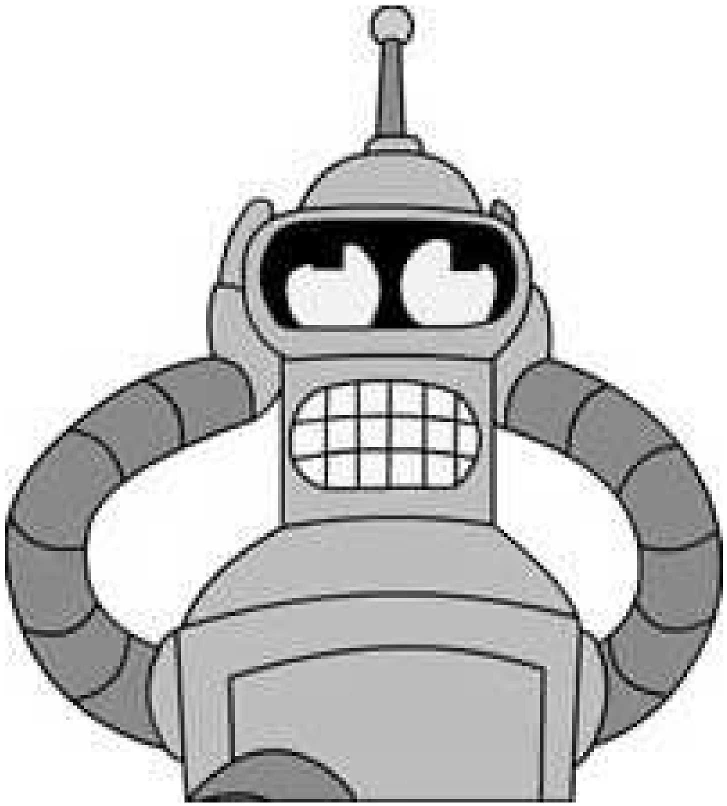
TT Enumeration Algorithm (cont.)

- Algorithm is sound and complete (proven by someone)
- But for a language with n propositions, requires searching 2^n models
- I.e. time complexity is $O(2^n)$

Summary

- Propositional Logic
  - Syntax
  - Semantics & Truth Tables
  - Natural Deduction
  - Inference with truth-table enumeration

Thank you!



“Ahhh, what an awful dream.
Ones and zeroes everywhere
... and I thought I saw a two!”