

# LOGICAL DEDUCTION IN AI

## PROPOSITIONAL LOGIC TO PREDICATE LOGIC



**Arijit Mondal & Partha P Chakrabarti**  
Indian Institute of Technology Kharagpur

# Deduction Using Propositional Logic: Steps

Choice of Boolean Variables  $a, b, c, d, \dots$  which can take values true or false.  
 $\sim \neg \quad \vee (\text{OR}) \quad \wedge (\text{AND}) \quad \rightarrow a \rightarrow b \quad \text{or } a \vee b$

Boolean Formulae developed using well defined connectors  $\sim, \wedge, \vee, \rightarrow$ , etc, whose meaning (semantics) is given by their truth tables.

Codification of Sentences of the argument into Boolean Formulae.

Developing the Deduction Process as obtaining truth of a Combined Formula expressing the complete argument.

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various Interpretations.

Validity   Tautology   Satisfiability

# Deduction Using Propositional Logic: Example 1

Choice of Boolean Variables  $a, b, c, d$ ,  
... which can take values true or false.

Boolean Formulae developed using well  
defined connectors  $\sim, \wedge, \vee, \rightarrow$ , etc,  
whose meaning (semantics) is given by  
their truth tables.

Codification of Sentences of the  
argument into Boolean Formulae.

Developing the Deduction Process as  
obtaining truth of a Combined Formula  
expressing the complete argument.

Determining the Truth or Validity of the  
formula and thereby proving or  
disproving the argument and Analyzing  
its truth under various Interpretations.

If I am the President then I am well-known. I am  
the President. So I am well-known

Coding: Variables

$a$ : I am the President ✓

$b$ : I am well-known ✓

Coding the sentences:

$F1: a \rightarrow b$

$F2: a$

$G: b$

$\neg a \vee b$

$F1 \wedge F2 \rightarrow G$

The final formula for deduction:  $(F1 \wedge F2) \rightarrow G$ ,

that is:

$((a \xrightarrow{F1} b) \wedge a) \rightarrow b$

This formula is always  
TRUE Tautology

# Deduction Using Propositional Logic: Example 1

Boolean variables  $a, b, c, d, \dots$  which can take values true or false.

Boolean formulae developed using well defined connectors  $\sim, \wedge, \vee, \rightarrow$ , etc, whose meaning (semantics) is given by their truth tables.

Codification of sentences of the argument into Boolean Formulae.

Developing the Deduction Process as obtaining truth of a combined formula expressing the complete argument.

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various interpretations.

If I am the President then I am well-known. I am the President. So I am well-known

Coding: Variables

$a$ : I am the President

$b$ : I am well-known

Coding the sentences:

F1:  $a \rightarrow b$

F2:  $a$

G:  $b$

The final formula for deduction:  $(F1 \wedge F2) \rightarrow G$ , that is:  $((a \rightarrow b) \wedge a) \rightarrow b$

TRUTH TABLES

$$a \rightarrow b \quad \boxed{\neg a \vee b}$$

$$p \rightarrow q \quad \neg p \vee q$$

| $a$ | $b$ | $a \rightarrow b$ | $(a \rightarrow b) \wedge a$ | $((a \rightarrow b) \wedge a) \rightarrow b$ |
|-----|-----|-------------------|------------------------------|----------------------------------------------|
| T   | T   | T ✓               | T ✓                          | T ✓                                          |
| T   | F   | F ✓               | F ✓                          | T ✓                                          |
| F   | T   | T ✓               | F ✓                          | T ✓                                          |
| F   | F   | T ✓               | F ✓                          | T ✓                                          |

# Deduction Using Propositional Logic: Example 2

Boolean variables  $a, b, c, d, \dots$  which can take values true or false. ✓

Boolean formulae developed using well defined connectors  $\sim, \wedge, \vee, \rightarrow$ , etc, whose meaning (semantics) is given by their truth tables.

Codification of sentences of the argument into Boolean Formulae. ✓

Developing the Deduction Process as obtaining truth of a combined formula expressing the complete argument. ✓

Determining the Truth or Validity of the formula and thereby proving or disproving the argument and Analyzing its truth under various interpretations. ✓

If I am the President then I am well-known. I am not the President. So I am not well-known

Coding: Variables

$a$ : I am the President ✓

$b$ : I am well-known ✓

Coding the sentences:

$F1$ :  $a \rightarrow b$  ✓

$F2$ :  $\sim a$

$G$ :  $\sim b$

The final formula for deduction:  $(F1 \wedge F2) \rightarrow G$ , that is  $((a \rightarrow b) \wedge \sim a) \rightarrow \sim b$

$F1 \wedge F2 \rightarrow G$  is a tautology  
interpretation VALID

T F

| $a$ | $b$ | $a \rightarrow b$ | $(a \rightarrow b) \wedge \sim a$ | $((a \rightarrow b) \wedge \sim a) \rightarrow \sim b$ |
|-----|-----|-------------------|-----------------------------------|--------------------------------------------------------|
| T   | T   | T                 | F                                 | T                                                      |
| T   | F   | F                 | F                                 | T                                                      |
| F   | T   | T                 | T                                 | F                                                      |
| F   | F   | T                 | T                                 | T                                                      |

$\neg a = T$

# Insufficiency of Propositional Logic

Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.

*all some no*

No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy.

Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

# Predicate Logic

First Order Logic

Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.

No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

New Additions in Proposition (First Order Logic)

Variables, Constants, Predicate Symbols and Function Symbols

New Connectors:  $\exists$  (there exists),  $\forall$  (for all)

Some  $\exists$  ✓  
all  $\forall$  ✓

$\text{goes}(x, y)$ :  $x$  goes to  $y$   
arguments  
variables

predicate symbol

$b$  ✓  
contractor( $x$ ) dependable( $x$ )  
engineer( $x$ )

# Formulating Predicate Logic Statements

New Additions in Proposition (First Order Logic)

Variables, Constants, Predicate Symbols and New Connectors:  $\exists$  (there exists),  $\forall$  (for all)

Example 1:  $\rightarrow$

Wherever Mary goes, so does the Lamb. Mary goes to School. So the Lamb goes to School.

Predicate: goes(x,y) to represent x goes to y

New Connectors:  $\exists$  (there exists),  $\forall$  (for all)

F1:  $\forall x(\text{goes}(\text{Mary}, x) \rightarrow \text{goes}(\text{Lamb}, x))$

F2: goes(Mary, School)

G: goes(Lamb, School)

To prove:  $(F1 \wedge F2) \rightarrow G$  is always true VALID

Mary Lamb  
School

Example 2

No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

Predicates: contractor(x), dependable(x), engineer(x)

F1:  $\forall x(\text{contractor}(x) \rightarrow \sim \text{dependable}(x))$

[Alternative:  $\sim \exists x(\text{contractor}(x) \wedge \text{dependable}(x))$ ]

F2:  $\exists x(\text{engineer}(x) \wedge \text{contractor}(x))$

G:  $\exists x(\text{engineer}(x) \wedge \sim \text{dependable}(x))$

To prove:  $(F1 \wedge F2) \rightarrow G$  is always true

$\exists x (\text{engineer}(x) \rightarrow \text{contractor}(x))$

$F1 \wedge F2 \rightarrow G$   $\wedge \exists x \text{engineer}(x)$



Example 3: -

All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

$$\begin{aligned}
 & \text{graceful}(x) \quad \text{student}(x) \\
 & \quad \quad \quad \text{dancer}(x) \quad \text{Ayesha} \\
 F1: & \forall x \{ \text{dancer}(x) \rightarrow \text{graceful}(x) \} \\
 & \cancel{\forall x \{ \text{dancer}(x) \wedge \text{graceful}(x) \}} \\
 F2: & \text{student}(\text{Ayesha}) \\
 F3: & \text{dancer}(\text{Ayesha}) \\
 G: & \exists x \{ \text{student}(x) \wedge \text{graceful}(x) \} \\
 & [(F1 \wedge F2 \wedge F3) \rightarrow G]
 \end{aligned}$$

# More Examples

Example: 4

Every passenger is either in first class or second class. Each passenger is in second class if and only if the passenger is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.

$$\begin{aligned}
 & p(x), f(x), s(x), w(x) \rightarrow \text{wealthy} \\
 & \quad \quad \quad \downarrow \text{passenger} \quad \downarrow \text{first class} \quad \downarrow \text{second class} \\
 F1: & \forall x \{ p(x) \rightarrow (f(x) \vee s(x)) \} \\
 F1': & \forall x \{ p(x) \rightarrow \{ (f(x) \wedge \neg s(x)) \vee (\neg f(x) \wedge s(x)) \} \} \\
 F2: & \forall x \{ p(x) \rightarrow ((s(x) \rightarrow \neg w(x)) \wedge (\neg w(x) \rightarrow s(x))) \} \\
 F3: & \exists x \{ p(x) \wedge w(x) \} \\
 F4: & \neg (\forall x \{ p(x) \rightarrow w(x) \}) \\
 G: & \exists x \{ p(x) \wedge s(x) \} \\
 & (F1 \wedge F2 \wedge F3 \wedge F4) \rightarrow G
 \end{aligned}$$

# Thank you

to Propositional Predicate Logic

variables constants  
predicates

$\forall$  for all  $\exists$  there exists

- codification of sentences into formulae
- development of the combined formula
- VALID / SATISFIABLE etc