Unit-3
Propositional Logic and First Order Logic

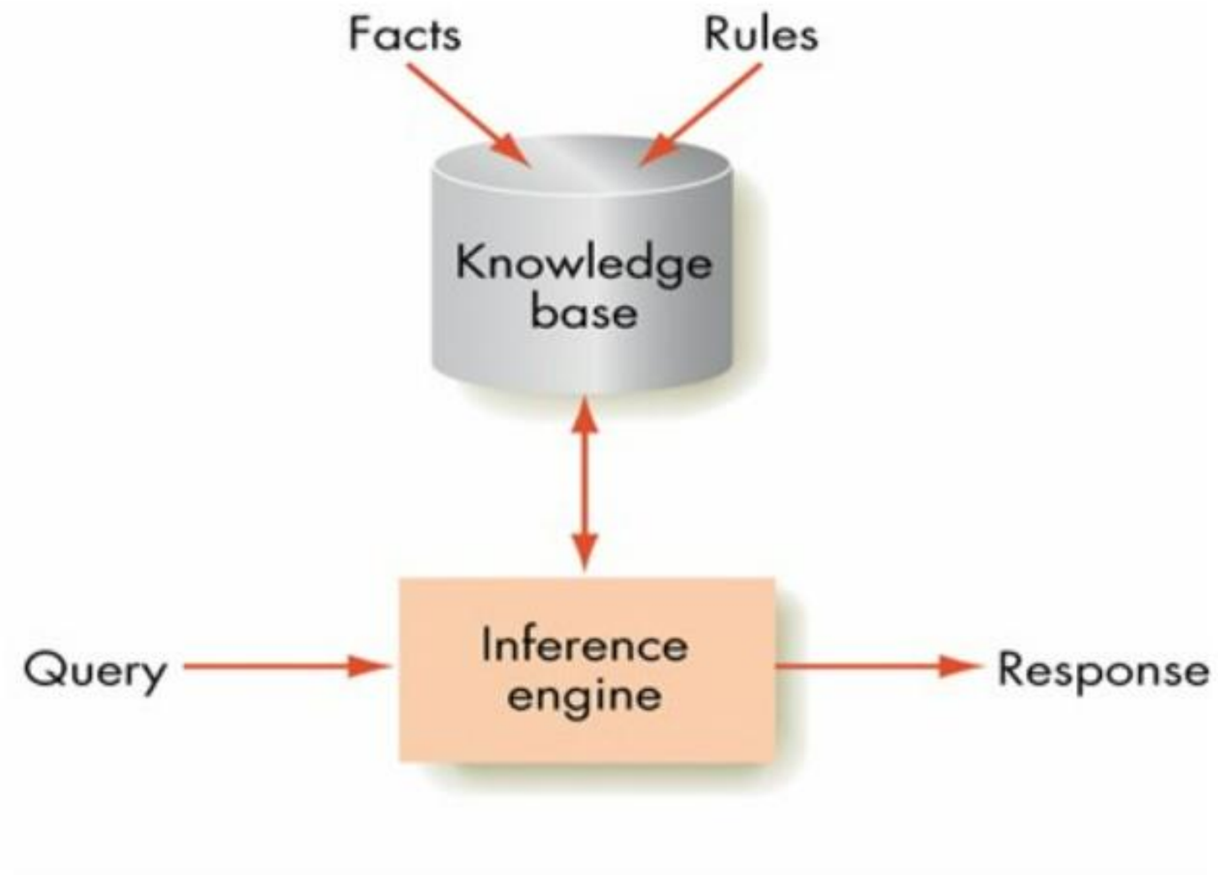
Knowledge-Base agent

- A knowledge-based agent can combine general knowledge with current percepts to infer hidden aspects of the current state prior to selecting actions.
- The central component of a knowledge-based agent is its knowledge base, or KB. Informally, a knowledge base is a set of sentences.
- Each sentence is expressed in a language called a knowledge representation language and some assertion about the world.

A knowledge-based agent is composed of:

1. Knowledge base: **domain-specific** content.

2. Inference mechanism: **domain-independent** algorithms.



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
- Deduce appropriate actions

Declarative approach to building an agent:

- Add new sentences: **Tell** it what it needs to know
- Query what is known: **Ask** itself what to do – answers should follow from the KB

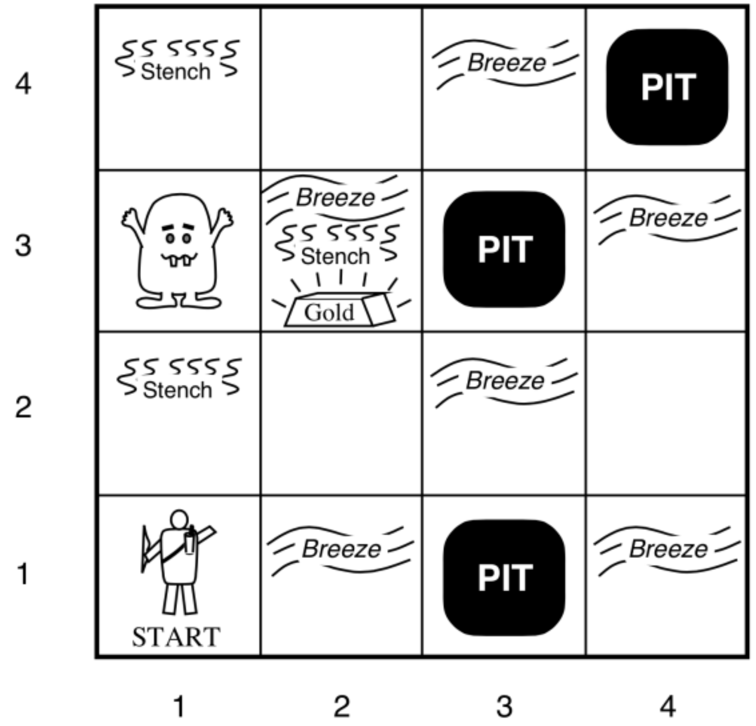
THE WUMPUS WORLD

- The wumpus world is a cave consisting of rooms connected by passageways.
- Lurking somewhere in the cave is the wumpus, a beast that eats anyone who enters its room.
- The wumpus can be shot by an agent, but the agent has only one arrow
- Some rooms contain bottomless pits that will trap anyone who wanders into these rooms (except for the wumpus, which is too big to fall in).

THE WUMPUS WORLD

- 4 X 4 grid of rooms
- Squares adjacent to Wumpus are smelly
- Squares adjacent to pit are breezy
- Glitter iff gold is in the same square
- Shooting kills Wumpus if you are facing it
- Wumpus emits a horrible scream when it is killed that can be heard anywhere
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square
- Releasing drops the gold in same square

Gregory Yob (1975)



THE WUMPUS WORLD PEAS

Performance measure:

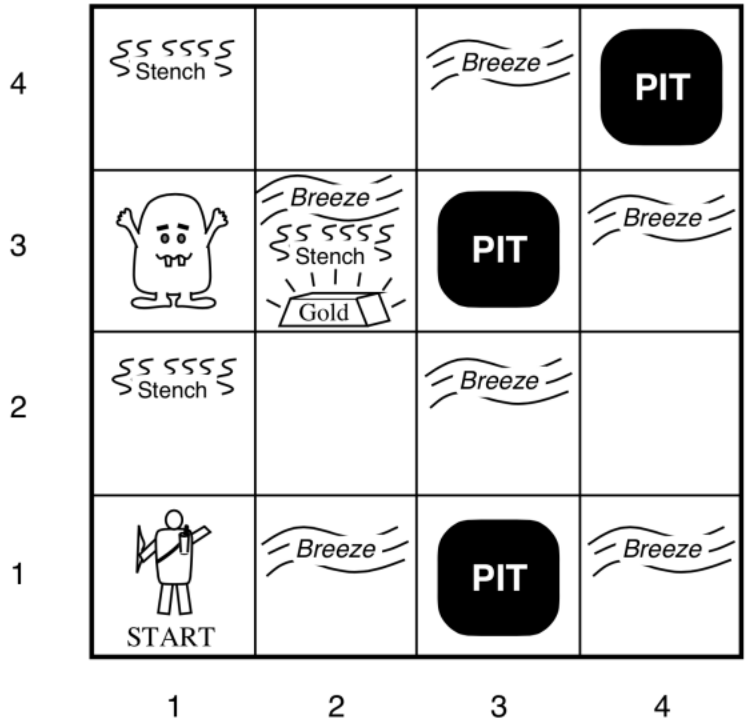
gold +1000, death (eaten or falling in a pit) -1000, -1 per action taken, -10 for using the arrow.

The games ends either when the agent dies or comes out of the cave.

Environment:

- 4 X 4 grid of rooms
- Agent starts in square [1,1] facing to the right
- Locations of the gold, and Wumpus are chosen randomly with a uniform distribution from all squares except [1,1]
- Each square other than the start can be a pit with probability of 0.2

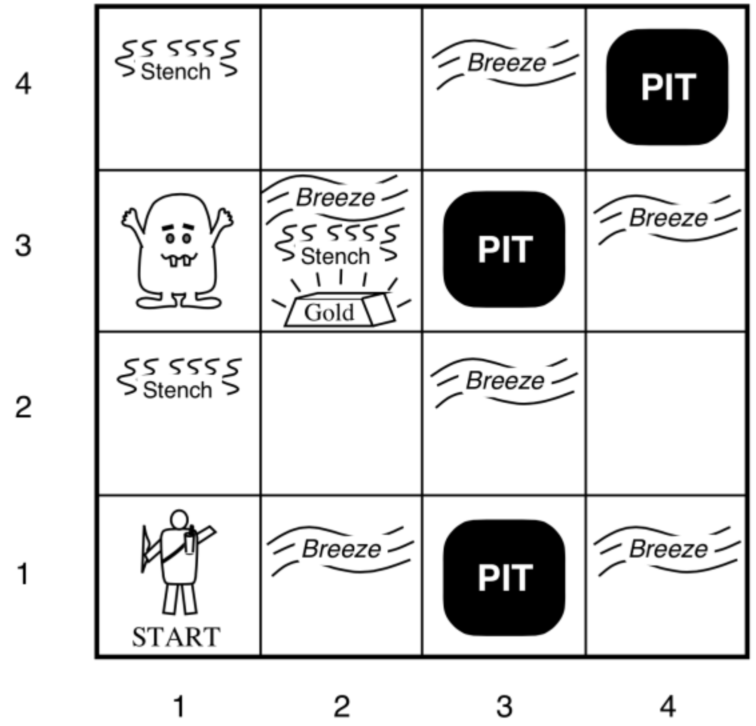
Gregory Yob (1975)



THE WUMPUS WORLD PEAS

Actuators:
Left turn, Right turn, Forward, Grab, Release, Shoot
Sensors:
Stench, Breeze, Glitter, Bump, Scream
Represented as a 5-element list
Example: [Stench, Breeze, None, None, None]

Gregory Yob (1975)



Wumpus World Characterization

Observable? No—only local perception

Deterministic? Yes—outcomes exactly specified

Episodic? No—sequential at the level of actions

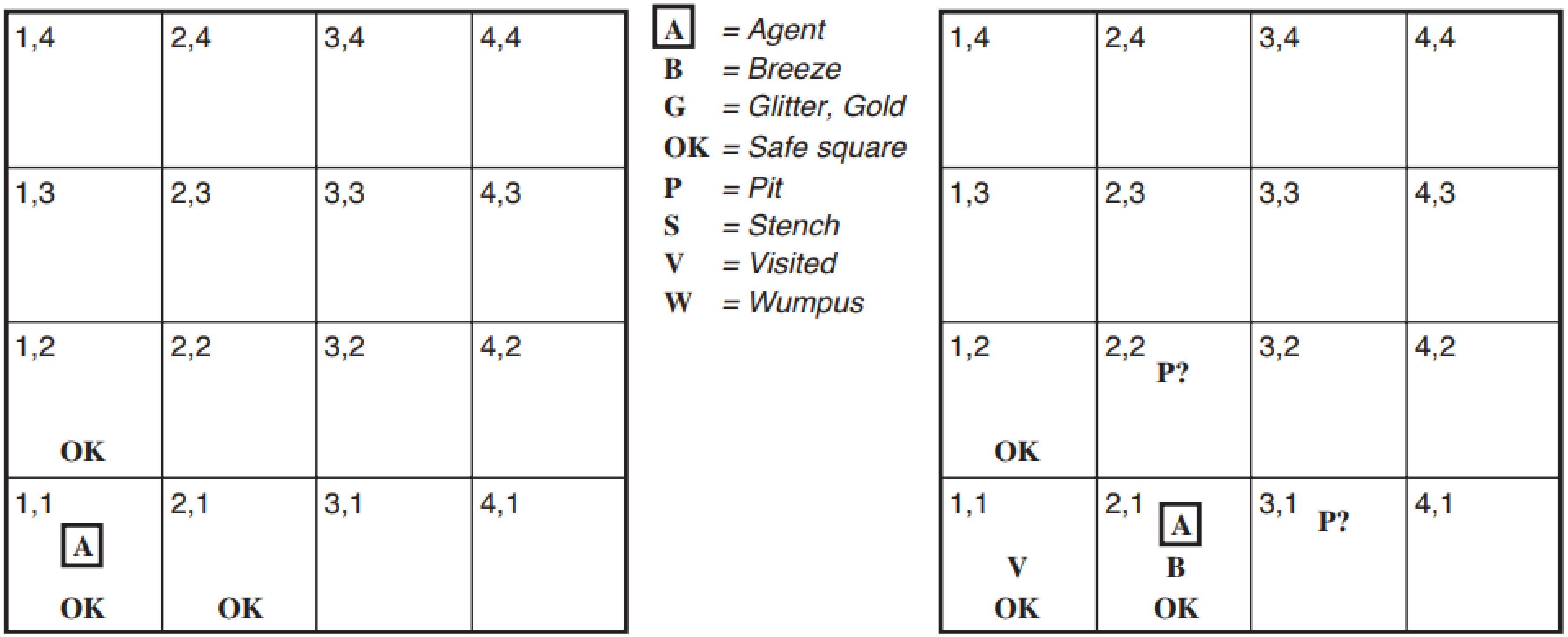
Static? Yes—Wumpus and Pits do not move

Discrete? Yes

Single-agent? Yes—Wumpus is essentially a natural feature

Exploring Wumpus world

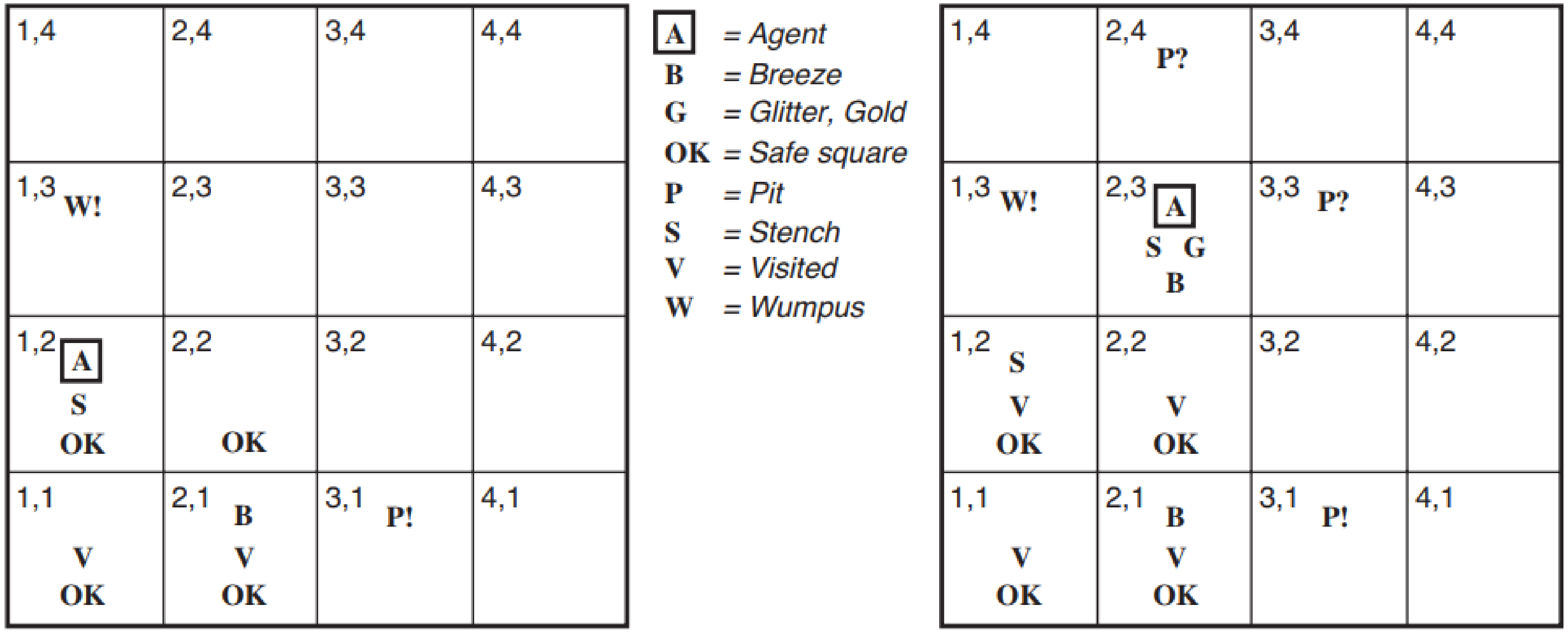
Agent's first steps:



(a) (b)

Exploring Wumpus world

Agent's later steps:



(a) (b)

Logic

- When most people say ‘logic’, they mean either *propositional logic* or *first-order predicate logic*

Any ‘formal system’ can be considered a logic if it has:

– a well-defined syntax;

– a well-defined semantics; and

– a well-defined proof-theory.

- *The syntax of a logic defines the syntactically acceptable objects of the language, which are properly called well-formed formulae (wff).*
- *The semantics of a logic associate each formula with a meaning.*
- Inference procedures (or a proof theory) define a means of deriving formulas from other formulas.

Propositional Logic (0th Order logic)

Definition: A proposition is a statement that can be either true or false; it must be one or the other, and it cannot be both.

Definition: A proposition is a declarative sentence that is either true (*denoted either T or 1*) or false (*denoted either F or 0*)

EXAMPLES. The following are propositions:

– the reactor is on;

– the wing-flaps are up;

– John Major is prime minister.

whereas the following are not:

– are you going out somewhere?

– Did John go to the store?

– x is greater than 2

– Look out!

Propositional Logic (0^{th} Order logic)

- Variables are used to represent propositions. The most common variables used are p, q, and r.

Logical Operators:

- Unary Operator negation: “not p”, $\neg p$.
- Binary Operators:

(a) conjunction: “p and q”, $p \wedge q$.

(b) disjunction: “p or q”, $p \vee q$.

(c) exclusive or: “exactly one of p or q”, “p xor q”, $p \oplus q$.

(d) implication: “if p then q”, $p \rightarrow q$.

(e) biconditional: “p if and only if q”, p iff q, $p \leftrightarrow q$.

Negation Operator, “not”, has symbol ¬

- The negation operator is a unary operator which, when applied to a proposition p, changes the truth value of p.
- if p is true, its negation is false

p	$\neg p$
T	F
F	T

Example:

p: This book is interesting.

¬p can be read as:

This book is not interesting.
This book is uninteresting.
It is not the case that this book is interesting.

- Another notation commonly used for the negation of p is ~ p.

Conjunction Operator, "and", has symbol ∧.

The conjunction operator is the binary operator which, when applied to two propositions *p* and *q*, yields the proposition "*p and q*", denoted $p \wedge q$. The conjunction $p \wedge q$ of *p* and *q* is the proposition that is true when both *p* and *q* are true and false otherwise.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction Operator: "or", has symbol ∨

- The disjunction operator is the binary operator which, when applied to two propositions p and q, yields the proposition "*p or q*", denoted $p \vee q$. The disjunction $p \vee q$ of p and q is the proposition that is true when either p is true, q is true, or both are true, and is false otherwise.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Exclusive Or. Exclusive Or Operator, “xor”, has symbol $\oplus$

- The *exclusive or* is the binary operator which, when applied to two propositions p and q yields the proposition “p xor q”, denoted p $\oplus$ q, which is true if exactly one of p or q is true, but not both. It is false if both are true or if both are false.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Implication Operator, "if…then…", has symbol →

We will write $p \rightarrow q$ for the conditional "if p then q"

In this conditional, the thing before the →(p in the example) is called the antecedent, premise, or hypothesis. The thing after the →(q in the example) is called the conclusion or consequence .

"If p then q" is false precisely when p is true but q is false.

Equivalent Forms of "If p then q":

- p implies q
- If p, q
- p only if q
- p is a sufficient condition for q
- q if p
- q whenever p
- q is a necessary condition for p

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication Operator, "if…then…", has symbol $\rightarrow$

For the compound statement $p \rightarrow q$:

- p is called the premise, hypothesis, or the antecedent.
- q is called the conclusion or consequent.
- $q \rightarrow p$ is the converse of $p \rightarrow q$.
- $\neg p \rightarrow \neg q$ is the inverse of $p \rightarrow q$.
- $\neg q \rightarrow \neg p$ is the contrapositive of $p \rightarrow q$.

Biconditional Operator, ”if and only if”, has symbol $\leftrightarrow$

- The biconditional statement is equivalent to $(p \rightarrow q) \wedge (q \rightarrow p)$. In other words, for $p \leftrightarrow q$ to be true we must have both p and q true or both false.

p	q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \wedge (q \rightarrow p) = p \leftrightarrow q$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

NAND and NOR Operators

- NAND (that is, not and), is a binary connective, written symbolically as p I q

p	q	$p \mid q$
T	T	F
T	F	T
F	T	T
F	F	T

- The NOR Operator (not or), which has symbol ↓ , written symbolically p ↓ q

p	q	$p \downarrow q$
T	T	F
T	F	F
F	T	F
F	F	T

Tautology and Contradiction

<u>Definitions:</u>

A compound proposition that is always true for all possible truth values of the propositions is called a tautology.

A compound proposition that is always false is called a contradiction.

A proposition that is neither a tautology nor contradiction is called a contingency.

Example: $p \vee \neg p$ is a tautology.

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

Tautology and Contradiction

<u>Definitions:</u>

A compound proposition that is always true for all possible truth values of the propositions is called a tautology.

A compound proposition that is always false is called a contradiction.

A proposition that is neither a tautology nor contradiction is called a contingency.

Example: $p \land \neg p$ is a contradiction.

p	$\neg p$	$p \land \neg p$
T	F	F
F	T	F

Logical equivalence

DeMorgan's Laws:

1) $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

2) $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$

p	q	$\neg p$	$\neg q$	$\neg(p \vee q)$	$\neg p \wedge \neg q$
T	T	F	F	**F**	**F**
T	F	F	T	**F**	**F**
F	T	T	F	**F**	**F**
F	F	T	T	**T**	**T**

Equivalence

Equivalent statements are important for logical reasoning since they can be substituted and can help us to:

(1) make a logical argument, and (2) infer new propositions

Example: $p \rightarrow q$ is equivalent to $\neg q \rightarrow \neg p$ (contrapositive)

p	q	$p \rightarrow q$	$\neg q \rightarrow \neg p$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Important logical equivalences

- Identity

$p \wedge T \Leftrightarrow p$

$p \vee F \Leftrightarrow p$

- Domination

$p \vee T \Leftrightarrow T$

$p \wedge F \Leftrightarrow F$

- Idempotent

$p \vee p \Leftrightarrow p$

$p \wedge p \Leftrightarrow p$

- Double negation

$\neg(\neg p) \Leftrightarrow p$

- Commutative

$p \vee q \Leftrightarrow q \vee p$

$p \wedge q \Leftrightarrow q\ p$

- Associative

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

$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

- Distributive

$P \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

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

- Other useful equivalences

$p \vee \neg p \Leftrightarrow T$

$p \wedge \neg p \Leftrightarrow F$

$p \rightarrow q \Leftrightarrow (\neg p \vee q)$

Show that $\neg[p \vee \neg(\neg q \vee \neg r)]$ is logically equivalent to $(p \vee q) \rightarrow \neg(p \vee r)$

$$
\begin{aligned}
\neg[p \vee \neg(\neg q \vee \neg r)] &\equiv \neg p \wedge (\neg q \vee \neg r) \\
&\equiv (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r) \\
&\equiv \neg(p \vee q) \vee (\neg p \wedge \neg r) \\
&\equiv (p \vee q) \rightarrow (\neg p \wedge \neg r) \\
&\equiv (p \vee q) \rightarrow \neg(p \vee r)
\end{aligned}
$$

Entailment ($|=$)

- Entailment means that one thing follows from another:
- Knowledge base KB entails sentence α if and only if:
  - ❖ α is true in all interpretations/worlds in which KB is true
  - ❖ if KB is true then α must be true

Write *KB* $|=$ α for KB entails α, So:

KB $|=$ α iff for every interpretation *I*, if *I* $|=$ *KB* then *I* $|=$ α.

Or

If *M*(α) is the set of all models of α, then *KB* $|=$ α iff $M(KB) \subseteq M(\alpha)$

Entailment (|=)

Entailment means that one thing follows from another:

Consider:

If it rains John takes an umbrella

If John takes an umbrella he doesn't get wet

If it doesn't rain then John doesn't get wet.

Show:

John doesn't get wet.

Propositions :

r: It rains

u: John takes an umbrella

w: John gets wet.

Query $\{r \rightarrow u, u \rightarrow \neg w, \neg r \rightarrow \neg w\} \models \neg w$

Now Prove that $\{r \rightarrow u, u \rightarrow \neg w, \neg r \rightarrow \neg w\} \models \neg w$

Solving logical inference problem

How to design the procedure that answers:

Three approaches:

1) Truth-table approach

2) Inference rules

3) Conversion to the inverse SAT problem

   – Resolution-refutation

Properties of inference solutions

1) Truth-table approach:
   - Blind
   - Exponential in the number of variables

2) Inference rules:
– More efficient
– Many inference rules to cover logic

3) Conversion to SAT - Resolution refutation:
– More efficient
– Sentences must be converted into CNF
– One rule – the resolution rule - is sufficient to perform all inferences

Truth table approach:

A two steps procedure:

1. Generate table for all possible interpretations

2. Check whether the sentence α evaluates to true whenever KB evaluates to true

Example: *KB= (A ∨C) ∧(B ∨~C)* $\alpha=(A \vee B)$

Problem with the truth table approach:

The truth table is exponential in the number of propositional symbols (we checked all assignments)

Observation: *KB is true only on a small subset interpretations*

A	B	C	$A \vee C$	$(B \vee \neg C)$	KB	α	
True	True	True	True	True	True	True	✓
True	True	False	True	True	True	True	✓
True	False	True	True	False	False	True	
True	False	False	True	True	True	True	✓
False	True	True	True	True	True	True	✓
False	True	False	False	True	False	True	
False	False	True	True	False	False	False	
False	False	False	False	True	False	False	

Solution:

Inference rules approach

-Start from entries for which KB is True.

-Generate new sentences from the existing ones

Inference rules approach:

Approach:

- Start from KB
- Infer new sentences that are true from existing KB sentences
- Repeat till α is proved (inferred true) or no more sentences can be proved

Rules:

(i) Equivalence rules: Logical equivalence rules

(ii) Inference rules:

Logical equivalences are discussed in the previous slides

Inference Rules

inference rule	tautology	name
$\begin{array}{rl} & p \\ & p \rightarrow q \\ \hline \therefore & q \end{array}$	$(p \wedge (p \rightarrow q)) \rightarrow q$	Modus ponens (mode that affirms)
$\begin{array}{rl} & \neg q \\ & p \rightarrow q \\ \hline \therefore & \neg p \end{array}$	$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$	Modus tollens (mode that denies)
$\begin{array}{rl} & p \rightarrow q \\ & q \rightarrow r \\ \hline \therefore & p \rightarrow r \end{array}$	$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$	hypothetical syllogism
$\begin{array}{rl} & p \vee q \\ & \neg p \\ \hline \therefore & q \end{array}$	$((p \vee q) \wedge (\neg p)) \rightarrow q$	disjunctive syllogism

Inference Rules

$\begin{array}{rl} & p \\ \hline \therefore & p \vee q \end{array}$	$p \rightarrow (p \vee q)$	addition
$\begin{array}{rl} & p \wedge q \\ \hline \therefore & p \end{array}$	$(p \wedge q) \rightarrow p$	simplification
$\begin{array}{rl} & p \\ & q \\ \hline \therefore & p \wedge q \end{array}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	conjunction
$\begin{array}{rl} & p \vee q \\ & \neg p \vee r \\ \hline \therefore & q \vee r \end{array}$	$((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$	resolution

Inference Rules

- Logical inference creates new sentences that logically follow from a set of sentences (KB)
- An inference rule is sound if every sentence X it produces from a KB logically follows from the KB
  - –i.e., inference rule creates no contradictions
- An inference rule is complete if it can produce every expression that logically follows from (is entailed by) the KB
  - –Note analogy to complete search algorithms

Inference rules approach.

Show that r follows from $p, p \to q$, and $q \to r$.

1.	p	Given
2.	$p \to q$	Given
3.	$q \to r$	Given
4.	q	MP: 1, 2
5.	r	MP: 3, 4

Modus Ponens $\dfrac{A; A \to B}{\therefore B}$

Inference rules approach.

Example:

Suppose $P \to Q$; $\neg P \to R$; $Q \to S$. Prove that $\neg R \to S$.

(1)	$P \to Q$	Premise
(2)	$\neg P \vee Q$	Logically equivalent to (1)
(3)	$\neg P \to R$	Premise
(4)	$P \vee R$	Logically equivalent to (3)
(5)	$Q \vee R$	Apply resolution rule to (2)(4)
(6)	$\neg R \to Q$	Logically equivalent to (5)
(7)	$Q \to S$	Premise
(8)	$\neg R \to S$	Apply HS rule to (6)(7)

Inference rule approach and Normal forms

Problems with inference rule approach:

-Too many different rules one can apply

-Many new sentence are just equivalent sentences

Question:

-Can we simplify inferences using one of the normal forms?

Normal forms:

1) Conjunctive normal form (CNF)

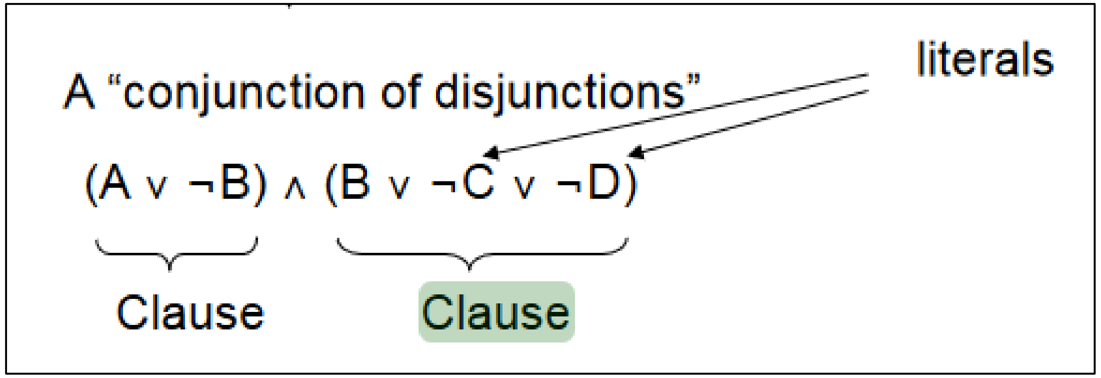
Conjunction of clauses (clauses include disjunctions of literals)

2) Disjunctive normal form (DNF)

Disjunction of terms (terms include conjunction of literals)

Normal forms

- A formula is in conjunctive normal form (CNF, clause normal form), if it is a conjunction of disjunctions of literals (or in other words, a conjunction of clauses).



- A formula is in disjunctive normal form (DNF), if it is a disjunction of conjunctions of literals.
- Checking the validity of CNF formulas or the unsatisfiability of DNF formulas is easy:
- A formula in CNF is valid, if and only if each of its disjunctions contains a pair of complementary literals P and ¬P .
- Conversely, a formula in DNF is unsatisfiable, if and only if each of its conjunctions contains a pair of complementary literals P and ¬P .

Conversion to a CNF

Assume: $\neg(A \Rightarrow B) \vee (C \Rightarrow A)$

1. Eliminate $\Rightarrow, \Leftrightarrow$

$$\neg(\neg A \vee B) \vee (\neg C \vee A)$$

2. Reduce the scope of signs through DeMorgan Laws and double negation

$$(A \wedge \neg B) \vee (\neg C \vee A)$$

3. Convert to CNF using the associative and distributive laws

$$(A \vee \neg C \vee A) \wedge (\neg B \vee \neg C \vee A)$$

and

$$(A \vee \neg C) \wedge (\neg B \vee \neg C \vee A)$$

Resolution algorithm

- Convert KB to the CNF form
- Now KB is in CNF
  - KB = AND of all the sentences in KB
  - KB sentence = clause = OR of literals
  - Literal = propositional symbol or its negation
- Find two clauses in KB, one of which contains a literal and the other its negation
- Cancel the literal and its negation
- Bundle everything else into a new clause
- Add the new clause to KB

Inference problem and satisfiability

Inference problem:

- we want to show that the sentence α is entailed by KB

Satisfiability:

- The sentence is satisfiable if there is some assignment (interpretation) under which the sentence evaluates to true

Connection:

$KB \models \alpha$ if and only if

$(KB \wedge \neg \alpha)$ is **unsatisfiable**

Consequences:

- inference problem is NP-complete
- programs for solving the SAT problem can be used to solve the inference problem

Inference problem and satisfiability

Inference problem:

- we want to show that the sentence α is entailed by KB

Satisfiability:

- The sentence is satisfiable if there is some assignment (interpretation) under which the sentence evaluates to true

Connection:

$$\boxed{\begin{array}{ll} KB \models \alpha & \text{if and only if} \\ (KB \wedge \neg \alpha) & \text{is } \textbf{unsatisfiable} \end{array}}$$

Consequences:

- inference problem is NP-complete
- programs for solving the SAT problem can be used to solve the inference problem

Resolution Rule

When applied directly to KB in CNF to infer α:

- **Incomplete:** repated application of the resolution rule to a KB in CNF may fail to derive new valid sentences

  Example:

  We know: $(A \wedge B)$ We want to show: $(A \vee B)$

  Resolution rule is incomplete

A trick to make things work:

- **proof by contradiction**
  - **Disproving:** $KB \wedge \neg \alpha$
  - **Proves the entailment** $KB \models \alpha$

  Resolution rule is refutation complete

Resolution Rule

A KB is a set of sentences all of which are true, i.e., a conjunction of sentences

- To use resolution, put KB into conjunctive normal form (CNF)
  - Each sentence is a disjunction of one or more literals (positive or negative atoms)
- Every KB can be put into CNF, it's just a matter of rewriting its sentences using standard tautologies, e.g.:

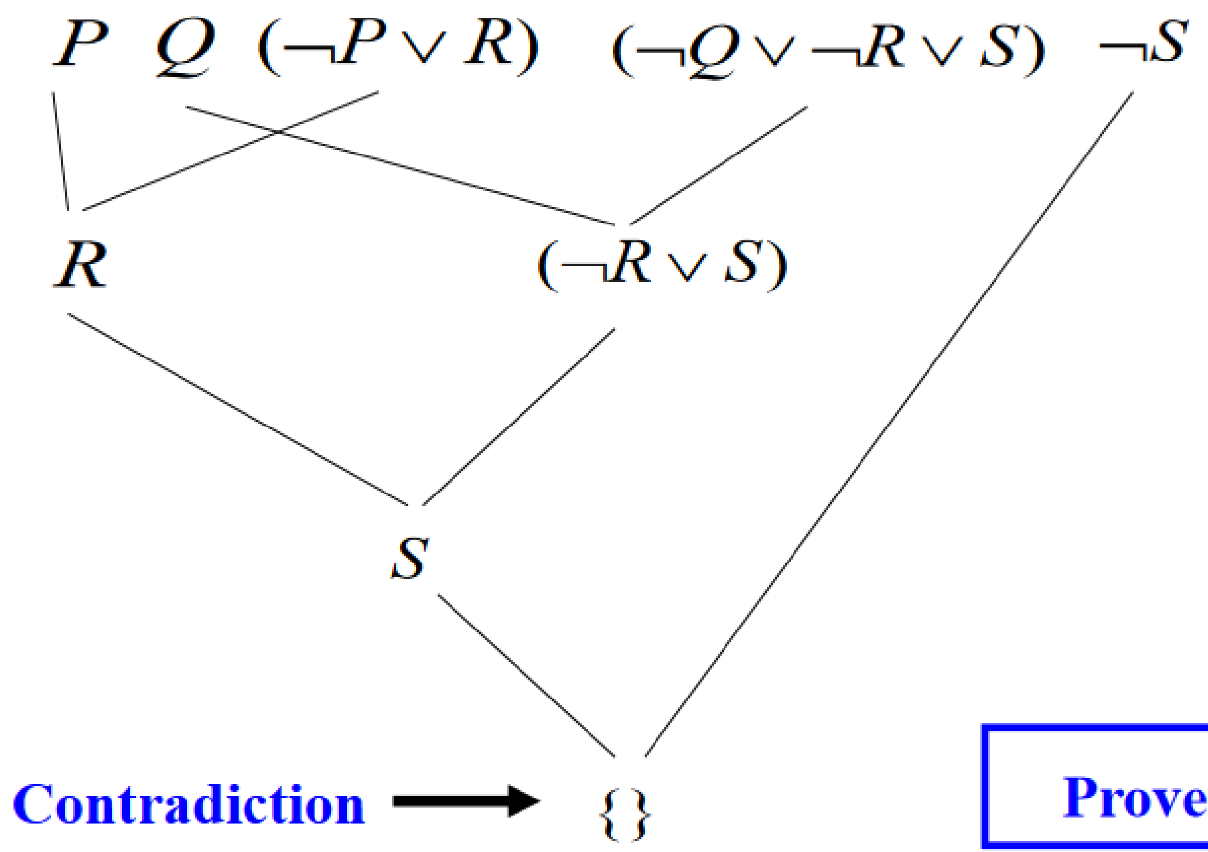
  $P \rightarrow Q \equiv {\sim}P \vee Q$

Resolution refutation

1. Add negation of goal to the KB
2. Convert all sentences in KB to CNF
3. Find all pairs of sentences in KB with complementary literals that have not yet been resolved
4. If there are no pairs stop else resolve each pair, by adding the result to the KB and go to 2
5. If we derived an empty clause (i.e., a contradiction) then the conclusion follows from the KB
6. If we did not, the conclusion cannot be proved from the KB

Example: Resolution

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ $\qquad \alpha = S$



Proved: S

Properties of inference solutions

- Truth-table approach
  - Blind
  - Exponential in the number of variables
- Inference rules
  - More efficient
  - Many inference rules to cover logic
- Conversion to SAT - Resolution refutation
  - More efficient
  - Sentences must be converted into CNF
  - One rule – the resolution rule - is sufficient to perform all inferences

Summary

Logical agents apply inference to a knowledge base to derive new information and make decisions.

Basic concepts of logic:

– syntax: formal structure of sentences

– semantics: truth of sentences wrt models

– entailment: necessary truth of one sentence given another

– inference: deriving sentences from other sentences

– soundness: derivations produce only entailed sentences

– completeness: derivations can produce all entailed sentences

Propositional logic limitations

(1) Statements that hold for many objects must be enumerated

Solution: make statements with variables

(2) Statements that define the property of the group of objects

Solution: make statements with quantifiers

(i) Universal quantifier –the property is satisfied by all members of the group

(ii) Existential quantifier – at least one member of the group satisfy the property

(3) Not expressive enough for most problems

First Order Logic (FOL)

- FOPL is also called predicate calculus , or Predicate logic
- Predicates are used to describe certain properties or relationships between individuals or objects.
- Quantifiers indicate how frequently a certain statement is true. Specifically, the *universal quantifier* is used to indicate that a statement is always true, whereas the *existential quantifier* indicates that a statement is sometimes true.
- Predicate Logic represented using *constants*, *variables* and *predicates*

First Order Logic (FOL)

- Constant –models a specific object

  Examples: “John”, “France”, “7”

- Variable – represents object of specific type (defined by the universe of discourse)

  Examples: x, y (universe of discourse can be people, students, numbers)

- Predicate - over one, two or many variables or constants.
  - Represents properties or relations among objects

  Examples: Red(car23), student(x), married(John, Ann)

Predicate Calculus: Syntax

The Domain (universe of discourse): The universe of discourse or domain is the collection of all persons, ideas, symbols, data structures, and so on, that affect the logical argument under consideration. The elements of the domain are called individuals or objects.

Predicates: *Properties* or *relations* among individuals or objects referred as predicates.

Variables: are frequently chosen from the end of the alphabet;

that is x, y and z

First Order Logic (FOL):Quantifiers

Rule of Inference	Name
$\dfrac{\forall x P(x)}{p(c) \text{ for an arbitrary element } c}$	Universal Instantiation
$\dfrac{P(c) \text{ for an arbitrary element } c}{\forall x P(x)}$	Universal generalization
$\dfrac{\exists x P(x)}{p(c) \text{ for some element } c}$	Existential Instantiation
$\dfrac{P(c) \text{ for some element } c}{\exists x P(x)}$	Existential generalization

First Order Logic (FOL):Quantifiers

Universal quantifier: The universal quantifier allows us to build formulae that are true for all objects.
Existential quantifier: The existential quantifier allows us to build formulae that are true for at least one object.

When $\forall x\ P(x)$ and $\exists x\ P(x)$ are true and false?

Statement	When true?	When false?
$\forall x\ P(x)$	$P(x)$ true for all x	There is an x where $P(x)$ is false.
$\exists x\ P(x)$	There is some x for which $P(x)$ is true.	$P(x)$ is false for all x.

Translation with quantifiers

Sentence: All KIIT students are smart.

Assume: the domain of discourse of x are KIIT students

Translation: $\forall x$ Smart(x)

Assume: the universe of discourse are students (all students):

$$\forall x \ at(x,KIIT) \rightarrow Smart(x)$$

Assume: the universe of discourse are people:

$$\forall x \ student(x) \wedge at(x, KIIT) \rightarrow Smart(x)$$

Translation with quantifiers

Sentence: Someone at KIIT is smart.

Assume: the domain of discourse are all KIIT affiliates

Translation: $\exists x\ Smart(x)$

Assume: the universe of discourse are people:

$$\exists x\ \ at(x\ ,\ KIIT) \wedge Smart(x)$$

Translation with quantifiers

Assume two predicates S(x) and P(x)

Universal statements typically tie with implications

- All S(x) is P(x)

  $\forall x\ (S(x) \rightarrow P(x))$

- No S(x) is P(x)

  $\forall x(S(x) \rightarrow \neg P(x))$

Existential statements typically tie with conjunctions

- Some S(x) is P(x)

  $\exists x\ (S(x) \wedge P(x))$

- Some S(x) is not P(x)

  $\exists x\ (S(x) \wedge \neg P(x))$

Free variable and bound variable

A variable is *bound* if it is under a quantifier with the same name (the occurrence of the variable close to the quantifier is called *binding* variable). If this is not the case, the variable is *free*.

Example, in the expression $\forall x(P(x) \rightarrow Q(x))$, the variable x appears three times and each time x is a bound variable.

Example, $\forall z(P(z) \wedge Q(x)) \vee \exists y Q(y)$. Here all occurrences of z and y are bound, Only one variable x is free.

Order of quantifiers

The order of nested quantifiers matters if quantifiers are of different type

- $\exists x \forall x\ P(x,y)$ is not the same as $\forall x\ \exists x\ P(x,y)$.

Example:

- Assume P(x,y) denotes "x loves y"
- Then: $\forall x\ \exists x\ L(x,y)$
- Translates to: Everybody loves somebody.
- And: $\exists x \forall x\ L(x,y)$
- Translates to: There is someone who is loved by everyone.

***The meaning of the two is different.

* *The order of nested quantifiers does NOT MATTER if quantifiers are of the SAME type

**The order of nested quantifiers MATTER if quantifiers are of the NOT same type

Inference in FOL : Example

Show that the premises:

- A student in Section A of the course has not read the book.
- Everyone in Section A of the course passed the first exam.

imply the conclusion

- Someone who passed the first exam has not read the book.

$A(x)$: "x is in Section A of the course"
$B(x)$: "x read the book"
$P(x)$: "x passed the first exam."

Hypotheses: $\exists x(A(x) \wedge \neg B(x))$ and $\forall x(A(x) \rightarrow P(x))$.
Conclusion: $\exists x(P(x) \wedge \neg B(x))$.

Inference in FOL:Example

Hypotheses: $\exists x(A(x) \land \neg B(x))$ and $\forall x(A(x) \to P(x))$.
Conclusion: $\exists x(P(x) \land \neg B(x))$.

Step	Reason
1. $\exists x(A(x) \land \neg B(x))$	Hypothesis
2. $A(a) \land \neg B(a)$	Existencial instantiation from (1)
3. $A(a)$	Simplification from (2)
4. $\forall x(A(x) \to P(x))$	Hypothesis
5. $A(a) \to P(a)$	Universal instantiation from (4)
6. $P(a)$	Modus ponens from (3) and (5)
7. $\neg B(a)$	Simplification from (2)
8. $P(a) \land \neg B(a)$	Conjunction from (6) and (7)
9. $\exists x(P(x) \land \neg B(x))$	Existential generalization from (8)

Inference in FOL: Example

Example: Suppose:

- all natural numbers are integers;
- there exists a natural number;

Prove that there exists an integer.

We can formalize this problem as follows. (Let the universe of discourse be all real numbers.)

$N(x)$: x is a natural number.
$I(x)$: x is an integer.
Premise: $\forall x\, (N(x) \rightarrow I(x)),\ \exists x\, N(x)$
Need to prove: $\exists x\, I(x)$

Inference in FOL: Example

(1)	$\exists x\, N(x)$	Premise
(2)	$N(c)$	Apply existential instantiation rule to (1)
(3)	$\forall x\, (N(x) \rightarrow I(x))$	Premise
(4)	$N(c) \rightarrow I(c)$	Apply universal instantiation rule to (3)
(5)	$I(c)$	Apply MP rule to (2)(4)
(6)	$\exists x\, I(x)$	Apply existential generalization rule to (5)

First order Predicate Logic

- Mary loves everyone. *$\forall x$ love (Mary, x), $\forall x$ (love (Mary, x)), ($\forall x$ love (Mary, x)), ($\forall x$ (love (Mary, x)))* *(All are correct)*
- Mary loves everyone: *$\forall x$ (person(x) → love (Mary, x))*
- Everyone loves himself. *$\forall x$ love (x, x)*
- Everyone loves everyone. *$\forall x \forall y$ love (x, y)*
- Everyone loves Mary. *$\forall x$ love (x, Mary)*
- Every student smiles. *$\forall x$ (student(x) → smile(x))*
- Every student except George smiles. *$\forall x$ ((student(x) & x ≠ George) → smile(x))*
- Everyone walks or talks. *$\forall x$ (walk (x) $\vee$ talk (x))*
- Every student who walks talks. *$\forall x$ (student(x) → (walk (x) $\vee$ talk (x)))*
- Everyone loves someone. *$\forall x \exists y$ love (x, y)* (or) *$\exists y \forall x$ love (x, y)*
- Someone loves everyone. *$\exists x \forall y$ love (x, y)* (or) *$\forall y \exists x$ love (x, y)*
- Someone walks and talks. *$\exists x$(walk (x) $\wedge$ talk (x))*

First order Predicate Logic

- Every person plays some game. .∀x ∃y Person(x) ⇒ (Game(y) ∧ Plays(x, y))
- All games are fun. ∀x Game(x) ⇒ Fun(x)
- For every game, there is a person that plays that game. ∀x ∃y [Game(x) ∧ Person(y)] ⇒ Plays(y, x)
- Every person plays every game. ∀x ∃y Game(x) ⇒ [Person(y) ∧ Plays(y, x)]
- Some person plays every game. ∃x ∀y Person(x) ∧ [Game(y) ⇒ Plays(x, y)]
- Some person plays some game. ∃x ∃y Person(x) ∧ Game(y) ∧ Plays(x, y)
- There is some person in Delhi who is smart. ∃x Person(x) ∧ In(x, Irvine) ∧ Smart(x)
- Every person in Delhi is smart. ∀x [Person(x) ∧ In(x, Irvine)] ⇒ Smart(x)

First order logic

(a) Marcus was a man.

(b) Marcus was a Roman.

(c) All men are people.

(d) Caesar was a ruler.

(e) All Romans were either loyal to Caesar or hated him (or both).

(f) Everyone is loyal to someone.

(g) People only try to assassinate rulers they are not loyal to.

(h) Marcus tried to assassinate Caesar.

(a) man(marcus)

(b) roman(marcus)

(c) $\forall$X. man(X) $\rightarrow$ person(X)

(d) ruler(caesar)

(e) $\forall$X. roman(x) $\rightarrow$ loyal(X,caesar) $\vee$ hate(X,caesar)

(f) $\forall$X$\exists$Y. loyal(X,Y)

(g) $\forall$X$\forall$Y. person(X) $\wedge$ ruler(Y) tryassasin(X,Y) $\rightarrow$ $\neg$loyal(X,Y)

(h) tryassasin(marcus,caesar)

First order logic

1) All apples are red

$\forall x\ (Apple(x) \Rightarrow Red(x))$

2) Every person has some person he loves

$\forall x\ \exists y\ Loves(x, y)$

3) There is a single person whom everybody loves.

$\exists y\ \forall x\ Loves(x, y)$

4) Every dog is owned by someone.

$\forall\ x(Dog(x)) \Rightarrow \exists\ y[Person(y) \wedge Owns(y,x)]$

5) John has a dog

$\exists x.\ Dog(x) \wedge Owns\ (John, x)$

6) Every DOG is a animal

$\forall x\ Dog(x) \rightarrow animal(x)$

7) Some dog is pet

$\exists x\ Dog(x) \rightarrow pet(x)$

8) Everyone loves somebody

$\forall x.\ \exists y.\ Loves(x,y)$

$\exists y.\ \forall x.\ Loves(x,y)$

9) Brothers are siblings

$\forall x,y\ Brother(x,y) \rightarrow sibling(x,y)$

$\forall x,y\ Brother(x,y) \leftrightarrow sibling(y,x)$

19) Ones mother is ones female parent

$\forall x,y\ Mother(x,y) \rightarrow (female(x) \wedge parent(x,y))$

$\forall x,y\ Mother(x,y) \leftrightarrow (female(x) \wedge parent(x,y))$

Types of Mathematical Logic

1) Propositional logic

Propositions are interpreted as true or false

Infer truth of new propositions

2) First order logic

Contains predicates, quantifiers and variables

- E.g. Philosopher(a) → Scholar(a)
- $\forall$x, King(x) $\land$ Greedy (x) → Evil (x)

Variables range over individuals (domain of discourse)

3) Second order logic

Quantify over predicates and over sets of variables

4) Temporal logic

Truths and relationships change and depend on time

5) Fuzzy logic

Uncertainty, contradictions