Unit - 4

Gary Marcus and Ernest Davis in piece for the *New York Times*:

“we need to stop building computer systems that merely get better and better at detecting statistical patterns in data sets—often using an approach known as ‘Deep Learning’—and start building computer systems that from the moment of their assembly innately grasp three basic concepts: time, space, and causality.”

Reasoning

- **Sentence 1:** I poured water from the bottle into the cup until it was full.
  Question: What was full, the bottle or the cup?
- **Sentence 2:** I poured water from the bottle into the cup until it was empty.
  Question: What was empty, the bottle or the cup?
- **Sentence 3:** Joe's uncle can still beat him at tennis, even though he is 30 years older.
  Question: Who is older, Joe or Joe's uncle?
- **Sentence 4:** Joe's uncle can still beat him at tennis, even though he is 30 years younger.
  Question: Who is younger, Joe or Joe's uncle?

Winograd schema challenge - https://en.wikipedia.org/wiki/Winograd_schema_challenge

Reasoning

- The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs.
- **Reasoning is a way to infer facts from existing data**.
- It is a general process of thinking rationally, to find valid conclusions.
- In artificial intelligence, the reasoning is essential so that the machine can also think rationally as a human brain, and can perform like a human.
- Basically, there are 3 parameters for reasoning, and they are:

  1. Rule

  2. Cause

  3. Effect

Reasoning in NLP

IN NLP there are primarily two subfields that study meaning, but they focus on different aspects of meaning in language use.

1. Semantics
2. Pragmatics

Semantics

Semantics is the study of **meaning** in language in an abstract, decontextualized sense. It deals with how words, phrases, and sentences convey meaning independent of the situation in which they are used. In other words, semantics is concerned with the **literal meaning** of expressions.

Key areas in semantics include:

- **Word meaning (lexical semantics)**: The meaning of individual words and their relationships to each other (e.g., synonyms, antonyms, hypernyms).
- **Compositional semantics**: How the meanings of words combine to form the meaning of larger structures, such as phrases or sentences.
- **Sentence meaning (truth conditions)**: The conditions under which a sentence can be considered true or false.
- **Ambiguity**: When a word or sentence has multiple meanings.

Semantics

Examples of **semantic** questions:

- What does the word "bank" mean (a financial institution or the side of a river)?
- What does the sentence "The cat is on the mat" mean, and under what conditions is it true?

Pragmatics

Pragmatics, on the other hand, deals with meaning in **context**. It studies how people use language in social interactions and how context influences the interpretation of utterances. Pragmatics considers **what speakers mean** by their statements in specific situations, including non-literal meanings such as implications, inferences, and presuppositions.

Key areas in pragmatics include:

- **Speech acts**: Actions performed by uttering words, such as requests, promises, commands, or questions.
- **Implicature**: What is suggested or implied by an utterance, even though it is not explicitly stated. (For example, "Can you pass the salt?" is not just a question but also a request.)
- **Deixis**: The way language points to or depends on context for interpretation, such as pronouns (I, you, he), time expressions (now, tomorrow), and place expressions (here, there).
- **Context**: The social, cultural, or situational circumstances in which a conversation occurs.

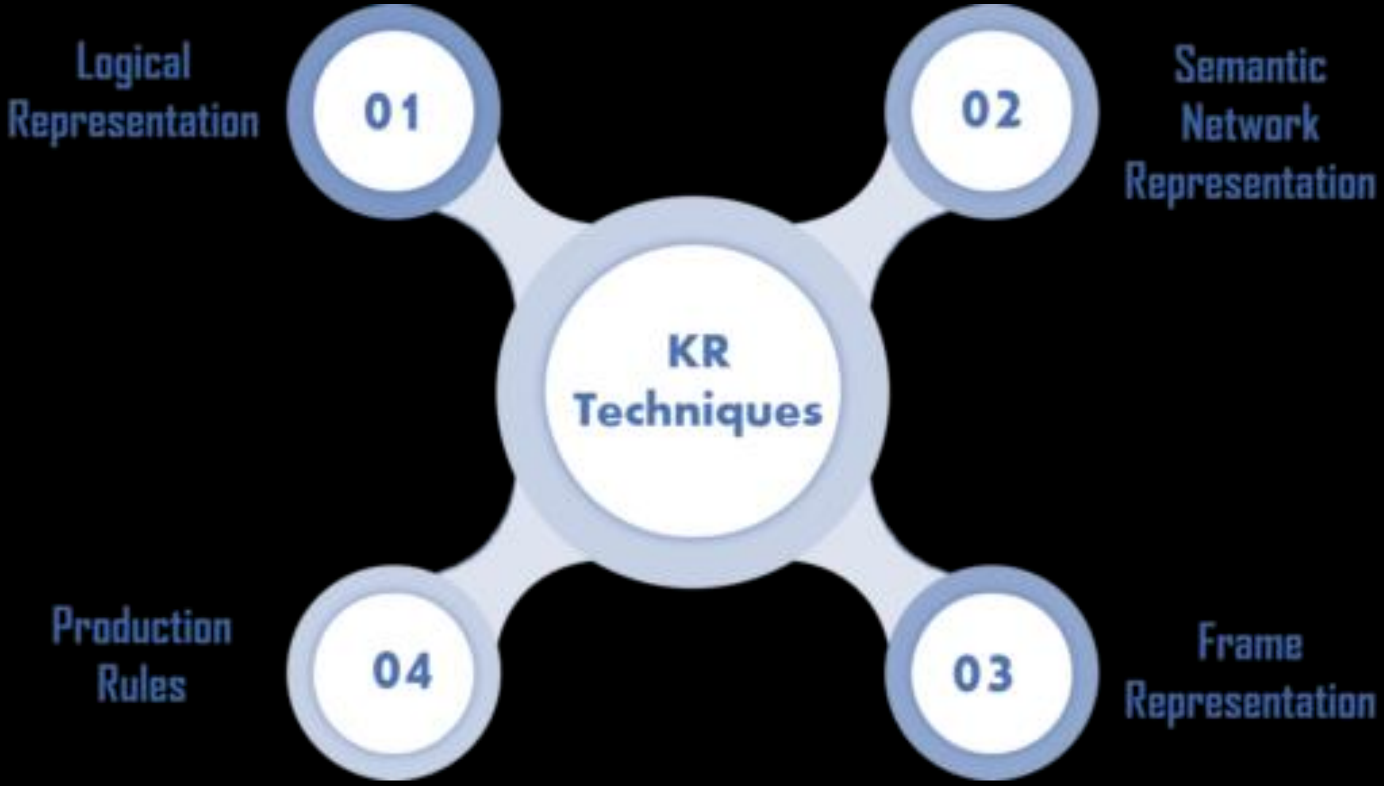
Pragmatics

- Examples of **pragmatic** questions:

- When someone says "It's cold in here," are they simply stating a fact or making a request to close the window?

- How do we understand the meaning of an utterance like "Can you close the window?" in a conversation, beyond the literal interpretation?

Four Techniques to knowledge representation

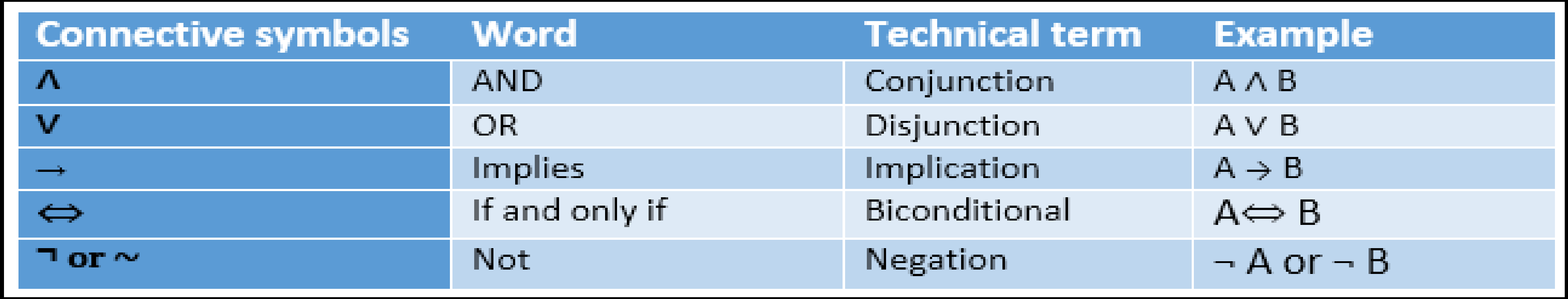


Logical Representation

- Logical representation is a language with some definite rules which deal with propositions and has no ambiguity in representation.
- It represents a conclusion based on various conditions and lays down some important communication rules.
- Also, it consists of precisely defined syntax and semantics which supports the sound inference.
- Each sentence can be translated into logics using syntax and semantics.

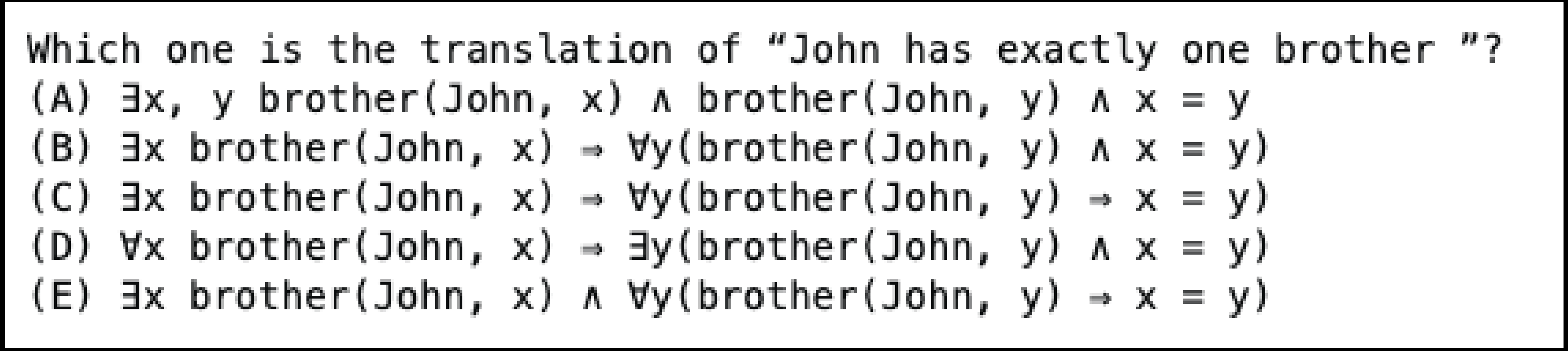
Logical Representation

1. Propositional logic



Connective symbols	Word	Technical term	Example
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

2. First order logic , etc.



Which one is the translation of “John has exactly one brother ”?

(A) $\exists x, y\ \text{brother}(John, x) \wedge \text{brother}(John, y) \wedge x = y$

(B) $\exists x\ \text{brother}(John, x) \rightarrow \forall y(\text{brother}(John, y) \wedge x = y)$

(C) $\exists x\ \text{brother}(John, x) \rightarrow \forall y(\text{brother}(John, y) \rightarrow x = y)$

(D) $\forall x\ \text{brother}(John, x) \rightarrow \exists y(\text{brother}(John, y) \wedge x = y)$

(E) $\exists x\ \text{brother}(John, x) \wedge \forall y(\text{brother}(John, y) \rightarrow x = y)$

Propositional Logic

- The simplest logic

- Definition: A proposition is a statement that is either true or false.

Propositional Logic

Examples:

1. Pitt is located in the Oakland section of Pittsburgh.

True

2. 5 + 2 = 8.

False

3. It is going to rain today.

(either T or F)

4. How are you?

a question is not a proposition

5. x + 5 = 3

since x is not specified, neither true nor false

6. 2 is a prime number.

True

7. She is very talented.

since she is not specified, neither true nor false

8. There are other life forms on other planets in the universe.

either T or F

Syntax : Propositional Logic

The syntax of propositional logic defines the allowable sentences for the knowledge representation.

There are two types of Propositions:

1. Atomic Propositions

2. Compound propositions

Syntax : Propositional Logic

Atomic Proposition:

- Atomic propositions are the simple propositions. It consists of a single proposition symbol.
- These are the sentences which must be either true or false.

Example-

- 1+1 is 2, it is an atomic proposition as it is a true fact.
- "The Sun is cold" is an atomic proposition as it is a false fact.

Syntax : Propositional Logic

Compound proposition:

- Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

Example-

- I am sleeping because lecture is boring.
- He is a doctor and his clinic is in Ahmedabad.
- There is a object in front so robot will take either left or right.

Logical Connectives

- **Logical connectives are used to connect two simpler propositions or representing a sentence logically.**
- **We can create compound propositions with the help of logical connectives.**

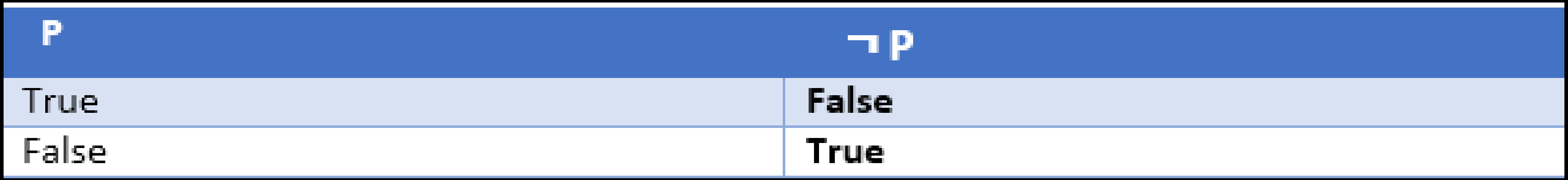
Logical Connectives

Negation:

- **A sentence such as ¬ P is called negation of P.**
- **A literal can be either Positive literal or negative literal.**

Example-

- The Sun is not cold
- P : The sun is cold
- **¬ P**



P	¬ P
True	**False**
False	**True**

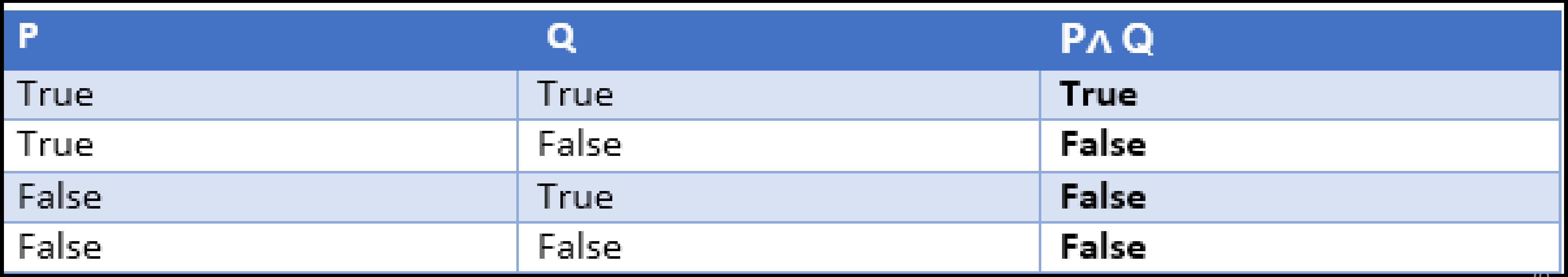
Logical Connectives

Conjunction:

- A sentence which has ∧ connective such as, P ∧ Q is called a conjunction.

Example: Prolog is simple and easy. It can be written as,

- P : Prolog is simple
- Q : Prolog is easy
- P ∧ Q.



P	Q	P∧ Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**False**

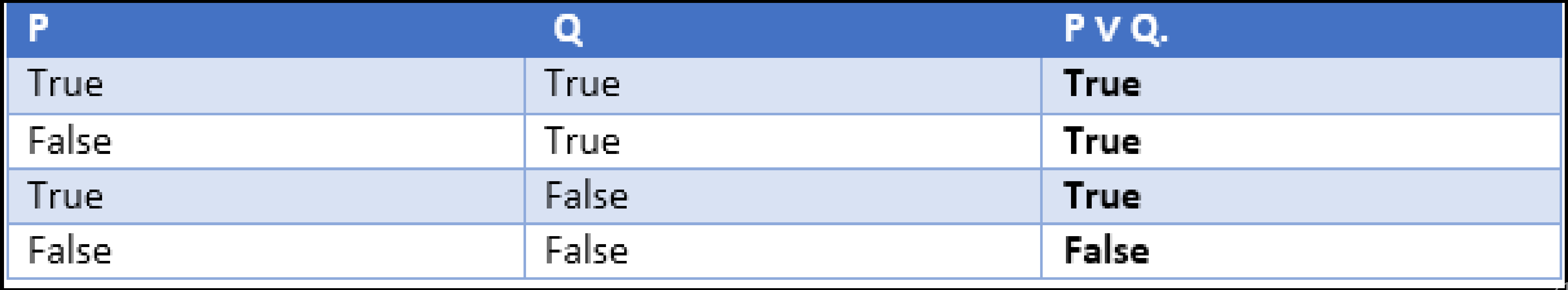
Logical Connectives

Disjunction:

- A sentence which has V connective, such as P V Q is called disjunction, where P and Q are the propositions.
- Example: “After 10th Student can take either A-Group or B-Group",

Example

- P= After 10th Student can take A-Group.
- Q= After 10th Student can take B-Group.



P	Q	P V Q.
True	True	**True**
False	True	**True**
True	False	**True**
False	False	**False**

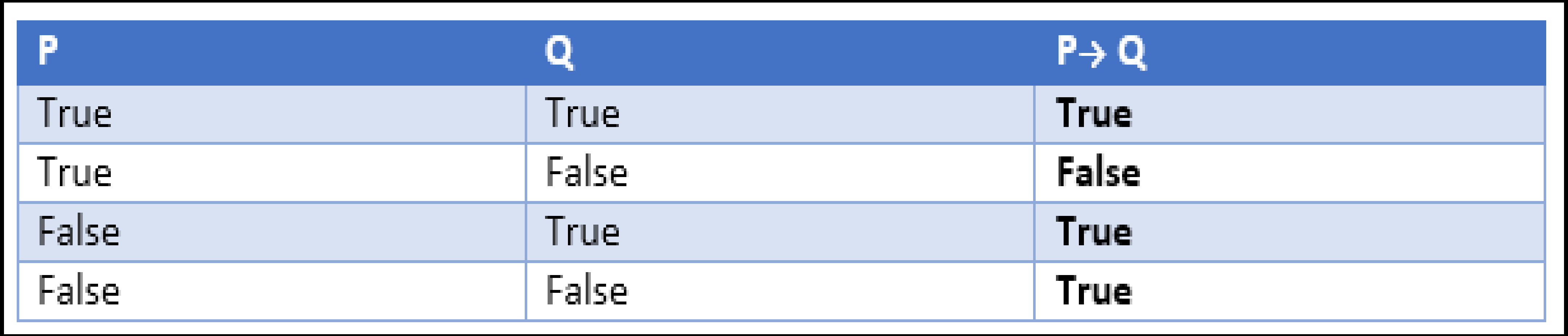
Logical Connectives

Implication:

- A sentence such as P → Q, is called an implication.
- Implications are also known as if-then rules.

Example - If it is raining, then the street is wet.

- P= It is raining
- Q= Street is wet
- It is represented as P → Q



P	Q	P→ Q
True	True	**True**
True	False	**False**
False	True	**True**
False	False	**True**

Logical Connectives

Implication:

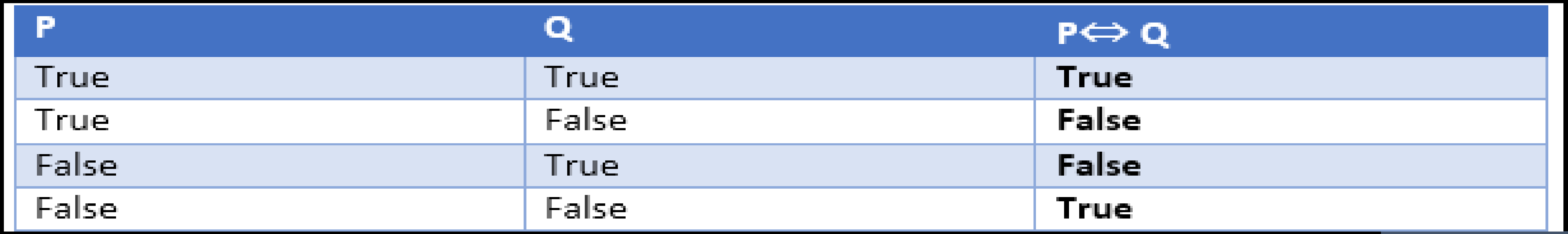
p → q may be interpreted as-

- If p then q
- p implies q
- q follows from p
- q if p
- q whenever p
- p only if q
- p is sufficient for q
- q is necessary for p
- q without p is possible and can exist
- p without q is impossible and can not exist

Logical Connectives

Bi-conditional:

- **A sentence such as P⇔ Q is a Bi-conditional sentence,**
- **Example If I am breathing, then I am alive**
- **P= I am breathing, Q= I am alive**
- **Tt can be represented as P ⇔ Q.**
- It is true when either both p and q are true or both p and q are false.
- It is false in all other cases.
- Bi-conditional is equivalent to EX-NOR Gate.



P	Q	P⇔ Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**True**

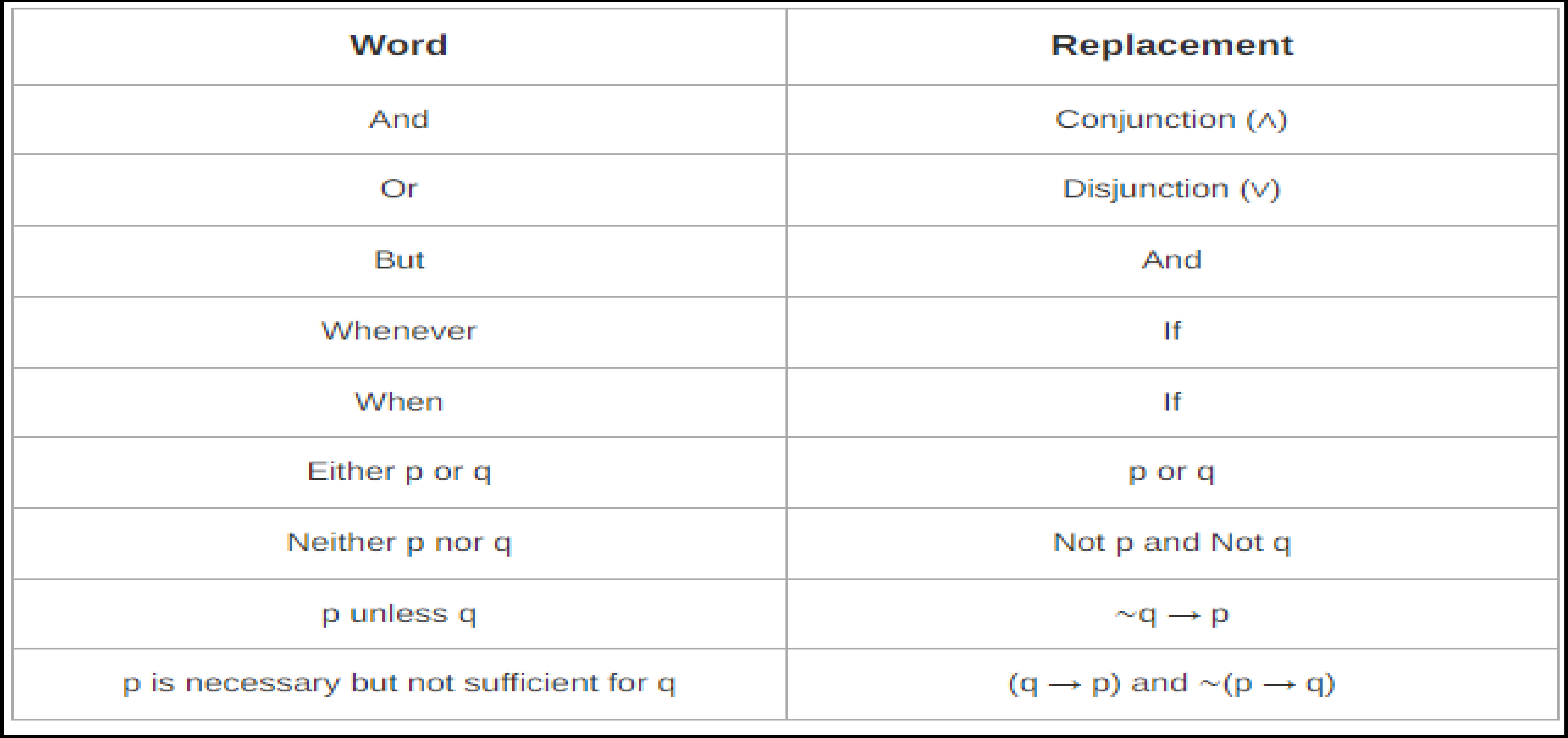
Logical Connectives

Bi-conditional:

$p \leftrightarrow q$ may be interpreted as-

- (If p then q) and (If q then p)
- p if and only if q
- q if and only if p
- (p if q) and (q if p)
- p is necessary and sufficient for q
- q is necessary and sufficient for p
- p and q are necessary and sufficient for each other
- p and q can not exist without each other
- Either p and q both exist or none of them exist
- p and q are equivalent
- $\sim p$ and $\sim q$ are equivalent

English Sentences To Propositional Logic



Word	Replacement
And	Conjunction (∧)
Or	Disjunction (∨)
But	And
Whenever	If
When	If
Either p or q	p or q
Neither p nor q	Not p and Not q
p unless q	$\sim q \rightarrow p$
p is necessary but not sufficient for q	$(q \rightarrow p)$ and $\sim(p \rightarrow q)$

Example

- The given sentence is- “If it rains, then I will stay at home.”
- This sentence is of the form- “If p then q”.
- So, the symbolic form is $p \rightarrow q$ where-
- p : It rains
- q : I will stay at home

Example

- We have-
- The given sentence is- “If a = b and b = c then a = c.”
- This sentence is of the form- “If p then q”.
- So, the symbolic form is $(p \wedge q) \rightarrow r$ where-
- $p : a = b$
- $q : b = c$
- $r : a = c$

Example

- We have-
- The given sentence is- “Neither it is hot nor cold today.”
- This sentence is of the form- “Neither p nor q”.
- “Neither p nor q” can be re-written as “Not p and Not q”.
- So, the symbolic form is $\sim p \land \sim q$ where-
- p : It is hot today
- q : It is cold today

Example

- We have-
- The given sentence is- “Birds fly if and only if sky is clear.”
- This sentence is of the form- “p if and only if q”.
- So, the symbolic form is $p \leftrightarrow q$ where-
- p : Birds fly
- q : Sky is clear

Example

- We have-
- The given sentence is- “It is hot or else it is both cold and cloudy.”
- It can be re-written as- “It is hot or it is both cold and cloudy.”
- So, the symbolic form is $p \vee (q \wedge r)$ where-
- p : It is hot
- q : It is cold
- r : It is cloudy

Example

- We have-
- The given sentence is- “We will leave whenever he comes.”
- We can replace “whenever” with “if”.
- Then, the sentence is- “We will leave if he comes.”
- This sentence is of the form- “q if p”.
-
- So, the symbolic form is $p \rightarrow q$ where-
- p : He comes
- q : We will leave

Example

- We have-
- The given sentence is- “Either today is Sunday or Monday.”
- It can be re-written as- “Today is Sunday or Monday.”
- So, the symbolic form is $p \vee q$ where-
- p : Today is Sunday
- q : Today is Monday

Example

- We have-

- The given sentence is- “Presence of cycle in a single instance RAG is a necessary and sufficient condition for deadlock.”

- This sentence is of the form- “p is necessary and sufficient for q”.

- So, the symbolic form is $p \leftrightarrow q$ where-

- p : Presence of cycle in a single instance RAG

- q : Presence of deadlock

Example

Consider the following two statements-

- S1 : Ticket is sufficient to enter movie theater.
- S2 : Ticket is necessary to enter movie theater.

Which of the statements is/ are logically correct?

(a) S1 is correct and S2 is incorrect.

(b) S1 is incorrect and S2 is correct.

(c) Both are correct.

(d) Both are incorrect.

Example

Consider the following two statements-

- S1 : Ticket is sufficient to enter movie theater.
- This statement is of the form- “p is sufficient for q” where-
- p : You have a ticket
- q : You can enter a movie theater
- So, the symbolic form is p → q
- For p → q to hold, its truth table must hold-

P (Ticket)	Q (Entry)	p → q (Ticket is sufficient for entry)
T	T	T
T	F	F
F	T	T
F	F	T

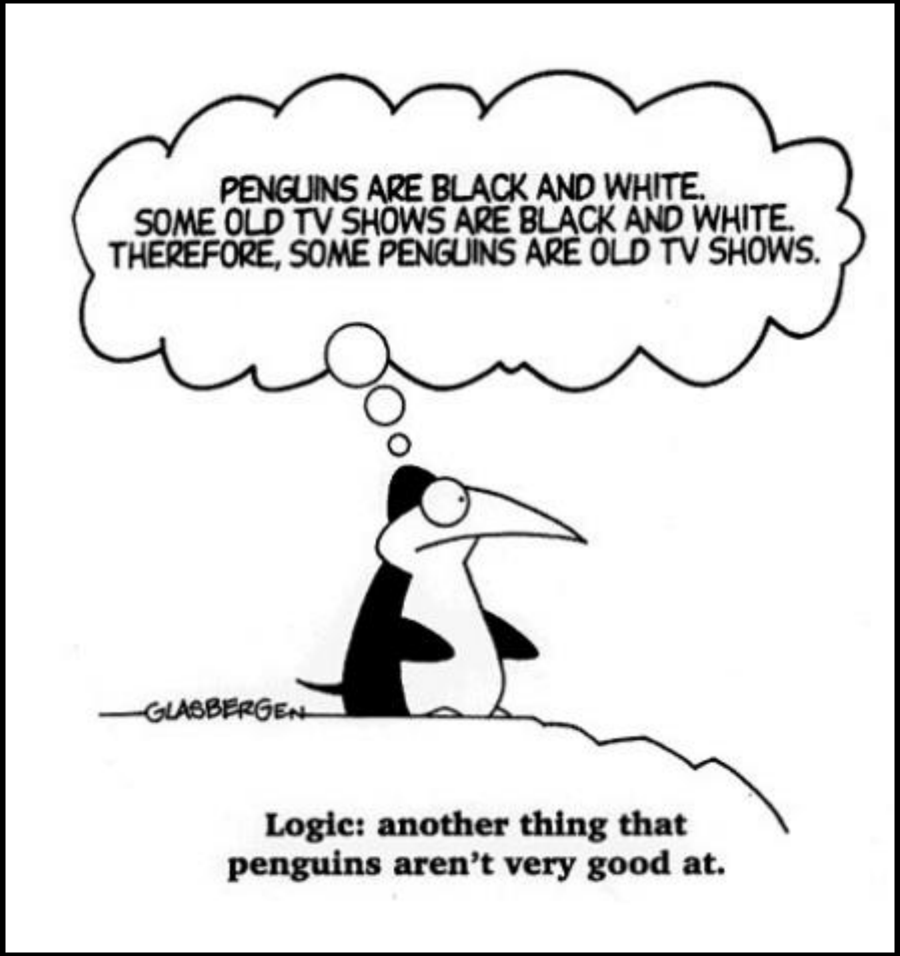
Example

Consider the following two statements-

- S2 : Ticket is Necessary To Enter Movie Theater-
- This statement is of the form- “q is necessary for p” where-
- p : You can enter a movie theater
- q : You have a ticket
- So, the symbolic form is $p \rightarrow q$
- For $p \rightarrow q$ to hold, its truth table must hold-

P (Entry)	**Q (Ticket)**	**$p \rightarrow q$ (Ticket is necessary for entry)**
F	F	T
F	T	T
T	F	F
T	T	T

First Order Logic


PENGUINS ARE BLACK AND WHITE.
SOME OLD TV SHOWS ARE BLACK AND WHITE.
THEREFORE, SOME PENGUINS ARE OLD TV SHOWS.
GLASBERGEN
Logic: another thing that penguins aren't very good at.

First Order Logic - Approach

- Adopt the foundation of propositional logic – a declarative, compositional semantics that is context-independent and unambiguous – but with more expressive power, borrowing ideas from natural language while avoiding its drawbacks

First Order Logic - Approach

Important elements in natural language:

- Objects (squares, pits)
- Relations among objects (is adjacent to, is bigger than) or unary relations or properties (is red, round)
- Functions (father of, best friend of)

First-order logic (FOL) is built around the above 3 elements

Propositional Logic vs. FOL

- Propositional logic assumes that there are facts that either hold or do not hold FOL assumes more: the world consists of objects with certain relations among them that do or do Not hold

Other special-purpose logics:

- Temporal logic: facts hold at particular times E.g., “I am always hungry”, “I will eventually be hungry”
- Higher-order logic: views the relations and functions in FOL as objects themselves
- Probability theory: facts / degree of belief [0, 1]
- Fuzzy logic: facts with degree of truth [0, 1] / known interval values

  E.g, “the temperature is very hot, hot, normal, cold, very cold”

First Order Logic

First-order logic is used to model the world in terms of

- objects which are things with individual identities

e.g., individual students, lecturers, companies, cars ...

- properties of objects that distinguish them from other objects

e.g., mortal, blue, oval, even, large, ...

- classes of objects (often defined by properties)

e.g., human, mammal, machine, ...

- relations that hold among objects

e.g., brother of, bigger than, outside, part of, has color, occurs after, owns, a member of, ...

- functions which are a subset of the relations in which there is only one ``value'' for any given ``input''.

e.g., father of, best friend, second half, one more than ...

First Order Logic - Syntax

A legitimate expression of predicate calculus is called well-formed formula (wff), or simply, sentence

Variables	**x, y, z, a, b,....**
Predicates	Brother, Father, >,....
Function	sqrt, LeftLegOf, ...
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀, ∃
Constant	1, 2, A, John, Mumbai, cat,....

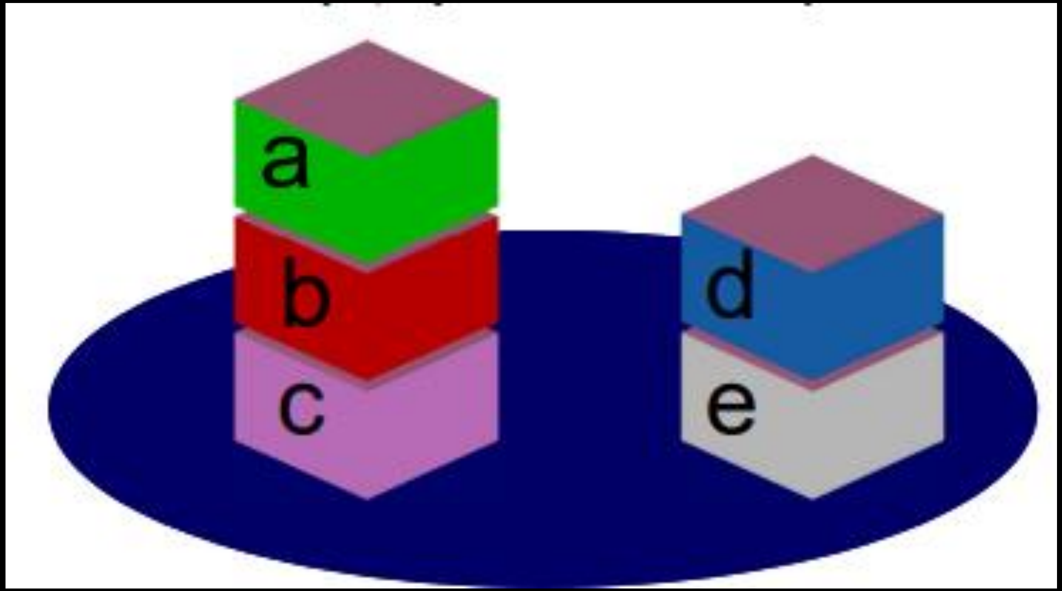
Relations (Predicates)

A relation is the set of tuples of objects

- Brotherhood relationship {<Richard, John>, <John, Richard>}
- Unary relation, binary relation, …

Example: set of blocks {a, b, c, d, e}

- The “On” relation includes:
- On = {<a,b>, <b,c>, <d,e>}
- the predicate On(A,B) can be interpreted as <a,b> Є On.



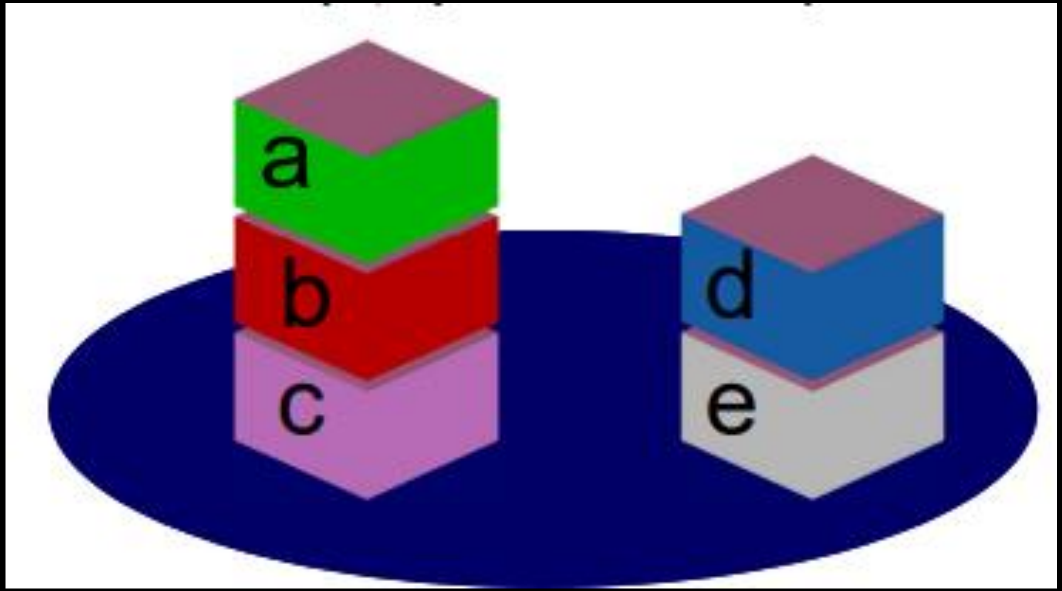
Functions

In English, we use “King John’s left leg” rather than giving a name to his leg, where we use “function symbol”

⍰ hat(c) = b

⍰ hat(b) = a

⍰ hat(d) is not defined



Terms and Atomic Sentences

- Atomic sentence = predicate (term1,...,termn) or term1 = term2
- Term = function (term1,...,termn) or constant or variable
- Atomic sentence states facts
- Term refers to an object

For example:

 Brother(KingJohn,RichardTheLionheart)

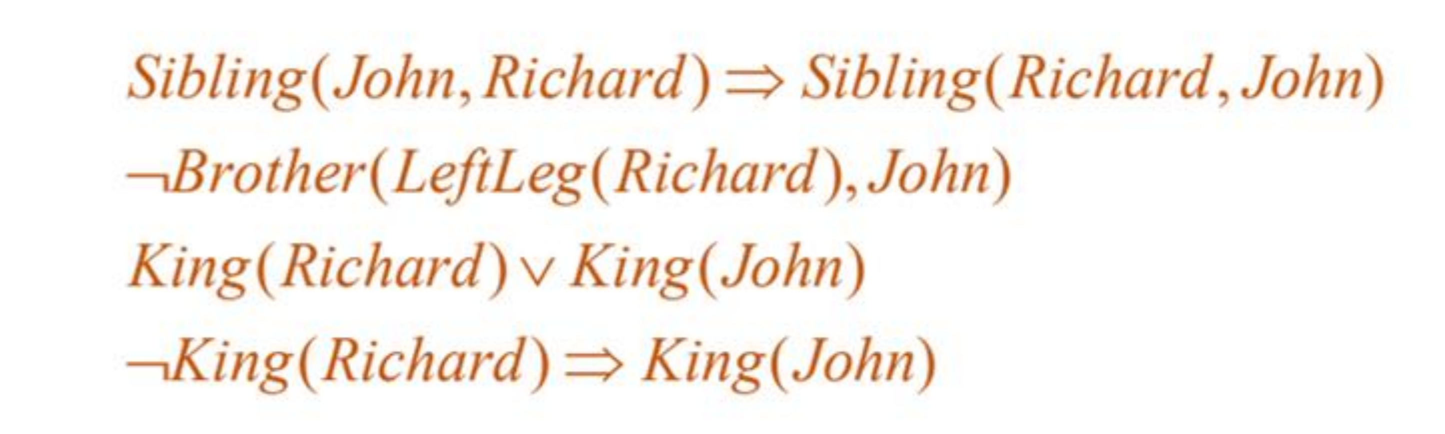
 Length(LeftLegOf(Richard))

 Married(Father(Richard), Mother(John))

Composite Sentences

- Complex sentences are made from atomic sentences using connectives Term refers to an object –
- ¬S, S1 ∧ S2, S1 ∨ S2, S1 ⇒ S2, S1 ⇔ S2

For example:



$$Sibling(John, Richard) \Rightarrow Sibling(Richard, John)$$

$$\neg Brother(LeftLeg(Richard), John)$$

$$King(Richard) \vee King(John)$$

$$\neg King(Richard) \Rightarrow King(John)$$

Quantifier

- A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.

1. Universal Quantifier, (for all, everyone, everything)
2. Existential quantifier, (for some, at least one).

Universal Quantifier

- Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.
- The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.
- Note: In universal quantifier we use implication "→".
- If x is a variable, then ∀x is read as:
- For all x
- For each x
- For every x.

Ex. All man drink coffee.

- ∀x man(x) → drink (x, coffee).

Existential Quantifier

- Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.
- It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.
- If x is a variable, then existential quantifier will be ∃x or ∃(x).

And it will be read as:

- There exists a 'x.'
- For some 'x.'
- For at least one 'x.„

Ex Some boys are intelligent.

- ∃x: boys(x) ∧ intelligent(x)

Translating English to FOL

- Every gardener likes the sun.

(∀x) gardener(x) => likes(x,Sun)

- Not Every gardener likes the sun.

~((∀x) gardener(x) => likes(x,Sun))

- You can fool some of the people all of the time.

(∃x)(∀t) person(x) ^ time(t) => can-be-fooled(x,t)

- You can fool all of the people some of the time.

(∀x)(∃t) person(x) ^ time(t) => can-be-fooled(x,t)

(the time people are fooled may be different)

Translating English to FOL

- You can fool all of the people at some time.

(∃t)(∀x) person(x) ^ time(t) => can-be-fooled(x,t)

(all people are fooled at the same time)

- You can not fool all of the people all of the time.

~((∀x)(∀t) person(x) ^ time(t) => can-be-fooled(x,t))

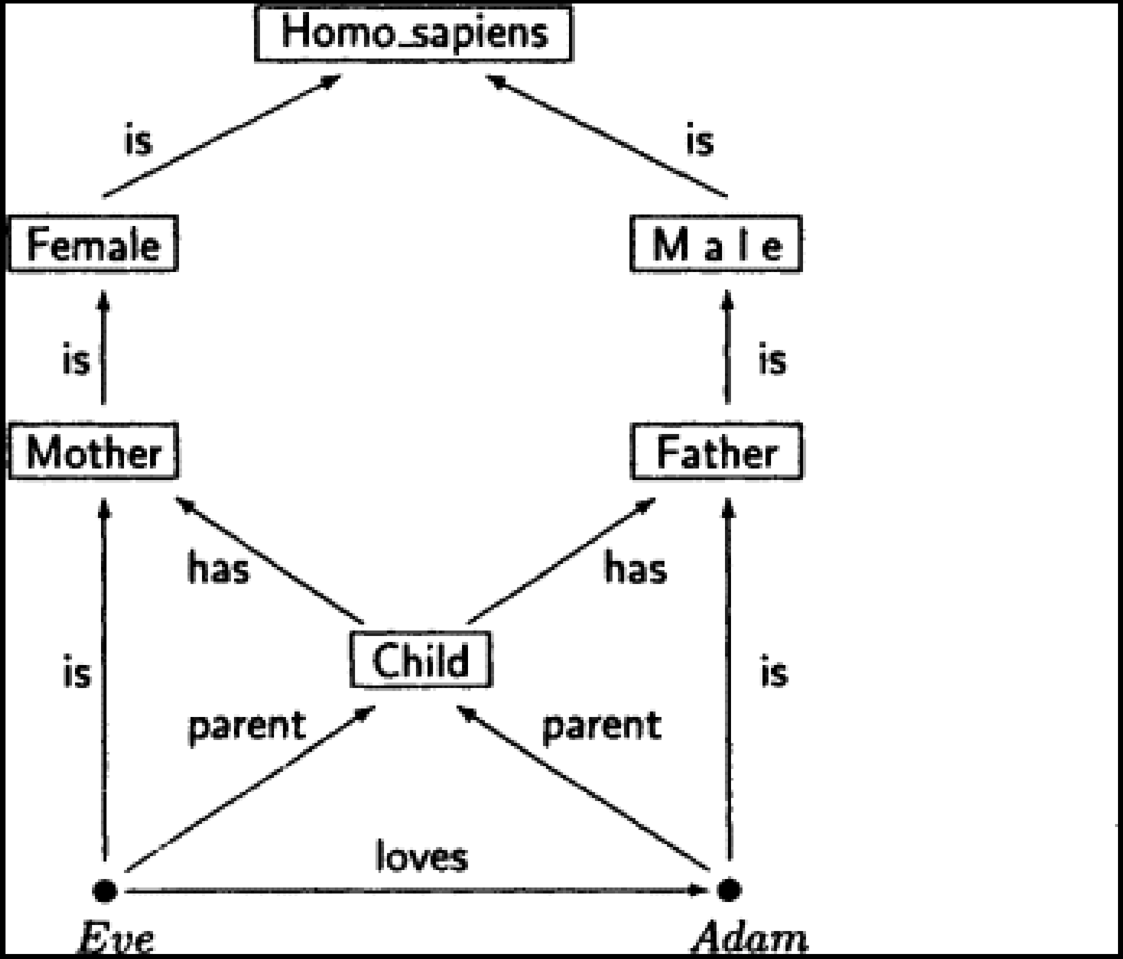
- Everyone is younger than his father

(∀x) person(x) => younger(x, father(x))

- All purple mushrooms are poisonous.

(∀x) (mushroom(x) ^ purple(x)) => poisonous(x)

Description Logic



Description Logic (DL)

- Description Logic (DL) is a family of formal knowledge representation languages used to describe and reason about the concepts and relationships in a particular domain.
- It is a type of formal logic that combines aspects of both first-order logic (FOL) and set theory, but it is specifically designed for structuring knowledge in domains that can be modeled using concepts, roles, and individuals.
- DL is primarily used in ontologies, particularly in the context of the Semantic Web, and for automated reasoning systems.

Description Logic (DL)

Key Features of Description Logics:

Concepts (Classes):

Concepts represent sets or collections of individuals (objects) in the domain. They are analogous to **sets** or **types** in traditional logic.

Examples: "Human," "Car," "Employee," etc.

Concepts are typically expressed using **constructs** that allow for defining their properties (e.g., intersection, union, negation, etc.).

Description Logic (DL)

Key Features of Description Logics:

Roles (Relations/Properties):

- **Roles define the relationships between individuals. They correspond to predicates or binary relations in first-order logic.**
- **Examples: "hasChild," "isMarriedTo," "worksAt," etc.**
- **A role can be thought of as a function that relates pairs of individuals (e.g., "John has a child named Alice").**

Individuals:

- **Individuals are the actual objects or entities in the domain of discourse.**
- **Examples: "Alice," "John," "Car123," etc.**

Description Logic (DL)

Key Features of Description Logics:

Axioms:

Description Logic allows the representation of statements or rules about concepts, roles, and individuals. These statements are called **axioms** and include:

TBox (Terminological Box): Contains general knowledge about concepts and roles, such as class hierarchies and constraints (e.g., "Human is a subclass of Animal").

ABox (Assertional Box): Contains specific facts or assertions about individuals, such as "John is a Human" or "Alice worksAt University".

Description Logic (DL)

Key Features of Description Logics:

Axioms:

Description Logic allows the representation of statements or rules about concepts, roles, and individuals. These statements are called **axioms** and include:

TBox (Terminological Box): Contains general knowledge about concepts and roles, such as class hierarchies and constraints (e.g., "Human is a subclass of Animal").

ABox (Assertional Box): Contains specific facts or assertions about individuals, such as "John is a Human" or "Alice worksAt University".

Description Logic (DL)

Common Constructs in Description Logics:

The power and expressiveness of a description logic system depend on the set of allowed **constructs** (logical operators) that it supports. Some of the common constructs include:

Conjunction (AND):

If C and D are concepts, then $C \sqcap D$ denotes the intersection of the concepts. It represents individuals that belong to both C and D.

Example: "Human Employee" represents individuals who are both humans and ⊓employees.

Disjunction (OR):

If C and D are concepts, then $C \sqcup D$ denotes the union of the concepts. It represents individuals that belong to either C or D.

Example: "Human Animal" represents individuals that are either humans or animals.

Description Logic (DL)

Negation (NOT):

If C is a concept, then ¬C denotes the complement of the concept. It represents individuals that do not belong to C.

Example: "¬Human" represents individuals that are not humans.

Existential Quantification ()∃:

If R is a role and C is a concept, then ∃ R.C represents individuals that are related by the role R to at least one individual that belongs to concept C.

Example: " ∃ hasChild.Human" represents individuals who have at least one child who is a human.\

Universal Quantification ()∀:

If R is a role and C is a concept, then ∀ R.C represents individuals that are related by the role R to only individuals that belong to concept C.

Example: " ∀ hasChild.Human" represents individuals whose children are all humans

Description Logic (DL)

Role Hierarchies:

Roles can be related to each other via inclusion relationships, where one role is a special case of another. For example, "hasSpouse" might be a more specific role of "hasPartner."

Qualified Cardinality Constraints:

This allows specifying how many individuals can be related to another individual via a role. For example, ≤1 hasChild means that an individual can have at most one child.

Description Logic (DL)

Role Hierarchies:

Roles can be related to each other via inclusion relationships, where one role is a special case of another. For example, "hasSpouse" might be a more specific role of "hasPartner."

Qualified Cardinality Constraints:

This allows specifying how many individuals can be related to another individual via a role. For example, ≤1 hasChild means that an individual can have at most one child.

Description Logic (DL)

Example of a Description Logic Knowledge Base:

Consider a small ontology in which we want to represent the concept of "Employee" and their relationships with companies and other roles.

Concepts:

Employee: A person who works for a company.

Manager: A special type of employee.

Company: An organization that employs individuals.

Roles:

worksFor: Represents the relationship between an employee and the company they work for.

manages: Represents the relationship between a manager and the employees they supervise.