Unit 4

Logic

A. Propositional logic

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- **A proposition is a declarative statement which is either true or false**.
- It is a technique of knowledge representation in **logical and mathematical form.**

- **Example**:

1. New Delhi is a capital of India
2. The square root of 4 is 2.
3. India will be superpower by 2030.
4. No, thank you.

Sentence 1 and 2 are propositions. They both are true.

Sentence 3 can not be said as ‘true’ or ‘false’, but not both.

Sentence 4 is an assertion, so we can not assign true or false with it.

Following are some basic facts about propositional logic:

- Propositional logic is also called **Boolean logic** as it works on **0 and 1**.
- In propositional logic, we use **symbolic variables** to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either **true or false, but it cannot be both**.
- Propositional logic consists of an **object**, **relations** or **function**, and **logical connectives**.
- These connectives are also called **logical operators**.
- The propositions and connectives are the basic elements of the propositional logic.
- **Connectives can be said as a logical operator which connects two sentences.**
- A proposition formula which is always true is called **tautology**, and it is also called a **valid sentence.**
- A proposition formula which is always false is called **Contradiction**.
- Statements which are questions, commands, or opinions are not propositions such as "**Where is Rohini**", "**How are you**", "**What is your name**", **are not propositions**.

Syntax of Propositional Calculus

- **Atomic Propositions**
- **Compound propositions**
- **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) **2+2 is 4**, it is an atomic proposition as it is a **true** fact.

2) "**The sun is cold**" is also a proposition as it is a **false** fact.

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

a) **“It is raining today, and street is wet.**

b) “Ajay is a doctor, **and** his clinic is in Mumbai”

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.
- There are mainly five connectives, which are given as follows:

1. **Negation :** A sentence such as $\neg$ P is called negation of P. A literal can be either Positive literal or negative literal.

2. **Conjunction :** A sentence which has **∧** connective such as, **P ∧ Q** is called a conjunction.
Example: Rohan is intelligent and hardworking. It can be written as,
P= Rohan is intelligent,
Q= Rohan is hardworking. → P∧ Q.

3. **Disjunction:** A sentence which has ∨ connective, such as **P ∨ Q**. is called disjunction, where P and Q are the propositions.
Example: "Ritika is a doctor or Engineer",
Here P= Ritika is Doctor.

Q= Ritika is an Engineer, so we can write it as **P ∨ Q**.

5. **Implication:** A sentence such as $P \rightarrow Q$, is called an implication. Implications are also known as if-then rules. It can be represented as

If it is raining, then the street is wet.

Let P= It is raining, and Q= Street is wet, so it is represented as $P \rightarrow Q$

6. **Biconditional :** A sentence such as **$P \Leftrightarrow Q$ is a Biconditional sentence, example If I am breathing, then I am alive**

P= I am breathing, Q= I am alive, it can be represented as $P \Leftrightarrow Q$.

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 $\neg B$

B. Semantics of Propositional Calculus

Like syntax, we will now define semantics for propositional calculus as follows:

- The truth assignment of negation : Where P is any propositional symbol is F, If the assignment to P is T; and T, if assignment to P is F.
- The truth assignment to Conjunction, is T only when both the conjuncts have the truth-value T; otherwise it is F.
- The truth assignment of disjunction, is F only when both the disjuncts have truth-value F; otherwise it is T.

- The truth assignment of implication, is F, only when the symbol before the implication, i.e. premise is T and the symbol after the implication, i.e. the consequent if F; otherwise it is T.
- The truth assignment of Biconditional, is T, only when symbols on both the sides are either T or F; otherwise it is F.

For Negation:

P	¬ P
True	**False**
False	**True**

For Conjunction:

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

For disjunction:

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

For Implication:

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

For Biconditional:

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

C. Well Formed Formula

- Propositional logic uses a symbolic “language” to represent the logical structure, or form, of a compound proposition.
- Like any language, this symbolic language has rules of syntax—grammatical rules for putting symbols together in the right way.
- **Any expression that obeys the syntactic rules of propositional logic is called a well-formed formula, or WFF.**

- the syntax of propositional logic is easy to learn. It has only three rules:

- Any capital letter by itself is a WFF.
- Any WFF can be prefixed with “~”. (The result will be a WFF too.)
- Any two WFFs can be put together with “^”, “∨”, “->”, or “<->” between them, enclosing the result in parentheses. (This will be a WFF too.)

WFF	non-WFF
A	A~
~A	(A)
~~A	(A ^)
(~A ^ B)	(A ^ B) v C)
((~A ^ B) v C)	

- Properties of Operators:
- **Commutativity:**
- $P \land Q = Q \land P$, or
- $P \lor Q = Q \lor P$.
- **Associativity**:
- $(P \land Q) \land R = P \land (Q \land R)$,
- $(P \lor Q) \lor R = P \lor (Q \lor R)$
- **Identity element:**
- $P \land True = P$,
- $P \lor True = True$.

- **Distributive**:
- $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$.
- $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.
- **DE Morgan's Law:**
- $\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$
- $\neg (P \vee Q) = (\neg P) \wedge (\neg Q)$.
- **Double-negation elimination:**
- $\neg (\neg P) = P$.

D. Properties of Statements

- A formula A in a language **Q** is **valid** if it is true for every interpretation of Q
- A formula A in a language Q is **satisfiable** if it is true for some interpretation of Q.
- A formula A of the language of arithmetic is **decidable** if it represents a decidable set, i.e. if there is an effective method which, given a substitution of the free variables of A, says that either the resulting instance of A is provable or its negation is.

Inferencing in Prepositional logic

- In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, so **generating the conclusions from evidence and facts is termed as Inference.**
- Inference rules are the templates for **generating valid arguments**.
- Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.
- In inference rules, **the implication among all the connectives plays an important role**.

Inferencing in Prepositional logic

- Following are some terminologies related to inference rules:
- **Implication**: It is one of the logical connectives which can be represented as $P \rightarrow Q$. It is a Boolean expression.
- **Converse**: The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as $Q \rightarrow P$.
- **Contrapositive**: The negation of converse is termed as contrapositive, and it can be represented as $\neg Q \rightarrow \neg P$.
- **Inverse**: The negation of implication is called inverse. It can be represented as $\neg P \rightarrow \neg Q$.

Limitations of Propositional logic:

- We **cannot represent relations like ALL, some, or none** with propositional logic. Example:
- All the girls are intelligent.
- Some apples are sweet.
- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties or logical relationships.

Predicate Logic (First Order Logic)

- In propositional logic, we can only represent the facts, which are either true or false.
- PL is not sufficient to represent the complex sentences or natural language statements.
- The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.

"Some humans are intelligent"

- To represent the above statements, PL logic is not sufficient, so we required some more powerful logic, such as first-order logic.

Predicate Logic (First Order Logic)

- First-order logic is another way of knowledge representation in artificial intelligence. **It is an extension to propositional logic.**
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- First-order logic is also known as **Predicate logic** or **First-order predicate logic**. First-order logic is a powerful language that develops information about the objects in a more easy way and can also **express the relationship between those objects.**
- First-order logic (like natural language) **does not only assume that the world contains facts like propositional logic** but also assumes the following things in the world:

Predicate Logic (First Order Logic)

- **Objects**: A, B, people, numbers, colors, wars, theories, squares
- **Relations**: **It can be unary relation such as: red, round, is adjacent, or n-any relation such as: the sister of, brother of, has color, comes between**
- **Function**: Father of, best friend, third inning of, end of, ...
- As a natural language, first-order logic also has two main parts:
- **Syntax**
- **Semantics**

Predicate Logic (First Order Logic)

- The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols.
- **Atomic sentences:**
- Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
- We can represent atomic sentences as
- Predicate (term1, term2, ..., term n/Object).
- Example: Ravi and Ajay are brothers: => Brothers(Ravi, Ajay).
- Chinky is a cat: => cat (Chinky).

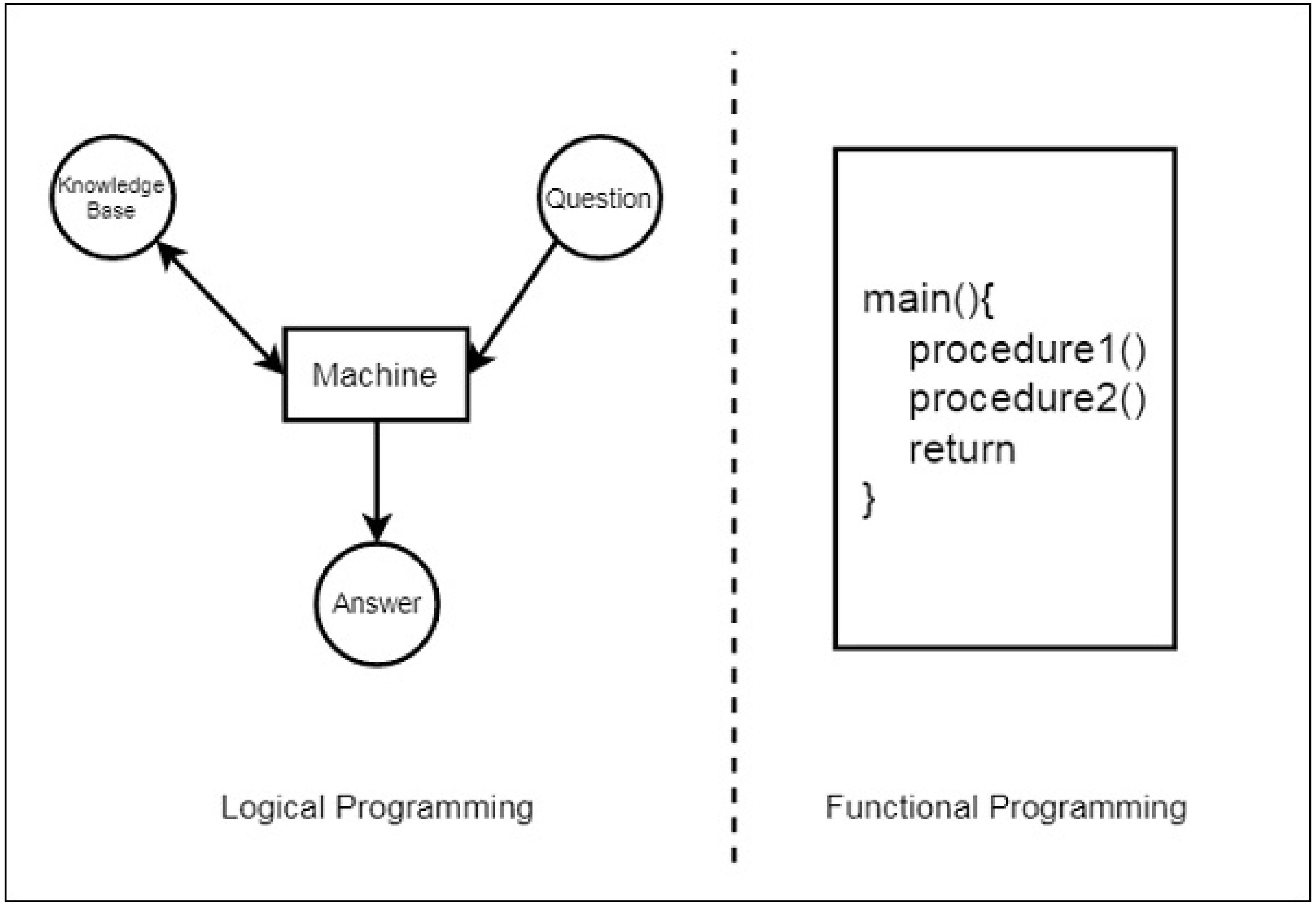
Prolog

- Prolog is a short form of **LOG**ical **PRO**gramming.
- **Prolog is a logic programming language.**
- Logic Programming is one of the Computer Programming Paradigm, in which th**e program statements express the facts and rules about different problems within a system of formal logic.**
- Prolog is intended primarily as a **declarative** programming language.
- In prolog, logic is expressed as relations (called as Facts and Rules).

Prolog

- To obtain the solution, **the user asks a question** rather than running a program. **When a user asks a question, then to determine the answer**, the run time system searches through the database of facts and rules.
- The first Prolog was '**Marseille Prolog**', which is based on work by Colmerauer.
- The major example of **fourth-generation programming language** was prolog.

Logic and Functional Programming



Prolog

- **Prolog language basically has three different elements −**
- **Facts** − The fact is predicate that is true, for example, if we say, “Tom is the son of Jack”, then this is a fact.
- **Rules** − Rules are extinctions of facts that contain conditional clauses. To satisfy a rule these conditions should be met. For example, if we define a rule as −
- grandfather(X, Y) :- father(X, Z), parent(Z, Y)
- This implies that for X to be the grandfather of Y, Z should be a parent of Y and X should be father of Z.
- **Questions** − to run a prolog program, we need **some questions, and those questions can be answered by the given facts and rules.**

Prolog

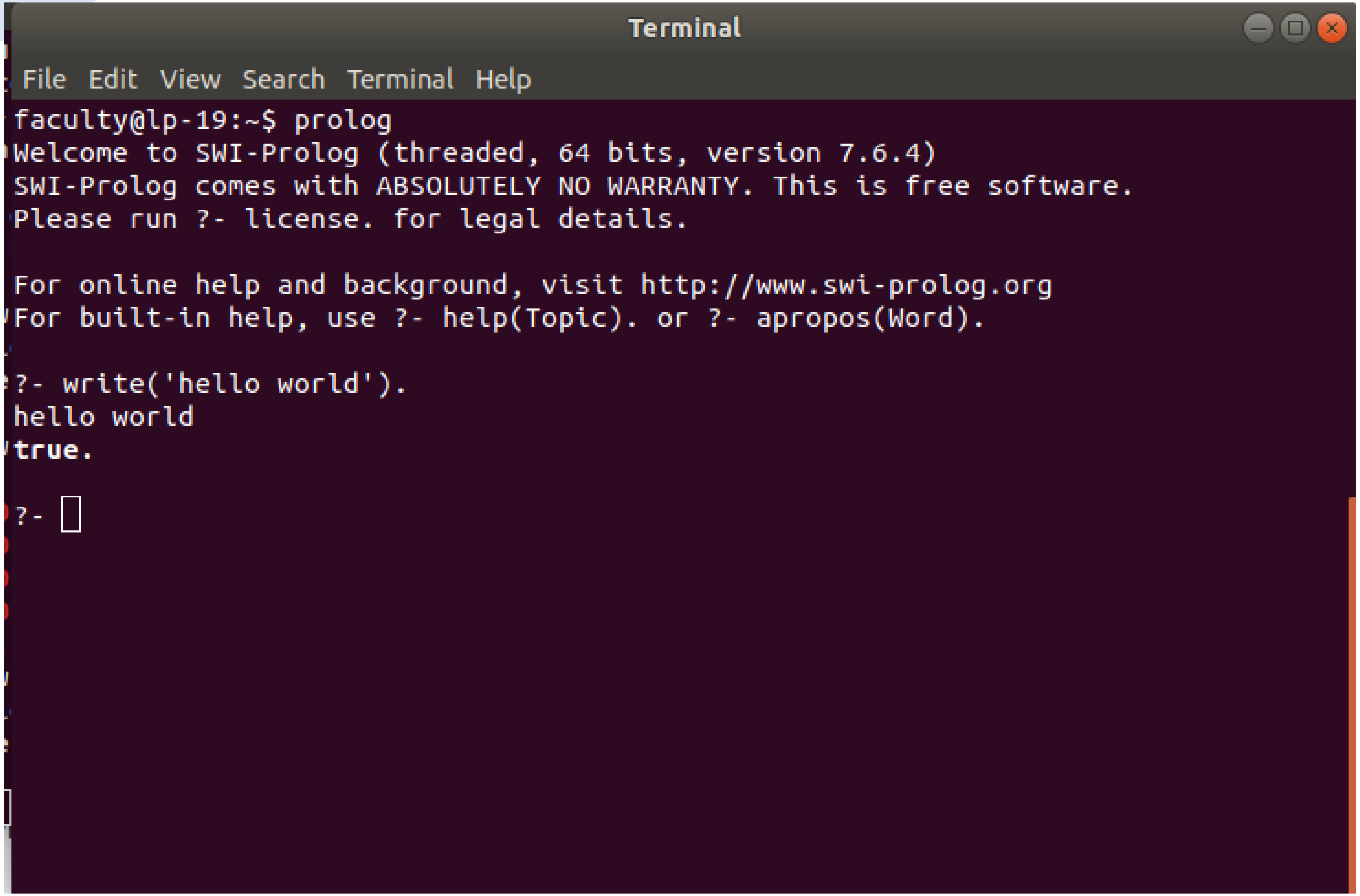
- Installation of Swi-prolog in Linux:

  - sudo apt-get install swi-prolog

- Prolog facts are expressed in definite pattern. Facts contain entities and their relation. Entities are written within the parenthesis separated by comma (,). Their relation is expressed at the start and outside the parenthesis. Every fact/rule ends with a dot (.). So, a typical prolog fact goes as follows :

- Format : relation(entity1, entity2, ...k'th entity).

Prolog


Terminal
File Edit View Search Terminal Help
faculty@lp-19:~$ prolog
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- write('hello world').
hello world
true.
?-

Prolog

- Example :

friends(raju, mahesh).
singer(sonu).
odd_number(5).

- Explanation :

These facts can be interpreted as :
raju and mahesh are friends.
sonu is a singer.
5 is an odd number.