Self-assignment: Propositional Logic and Predicate Logic

2024

Introduction to Propositional Logic

Propositional logic, also known as Boolean logic, deals with statements that can either be true or false. A statement is called a **proposition**, and it is represented by symbols like P, Q, or R.

0.1 Logical Connectives

Logical connectives are used to combine propositions to form new ones. The most common connectives are:

- **Negation** ($\neg P$): The negation of P is true when P is false.
- **Conjunction** ($P \wedge Q$): P and Q are true if both P and Q are true.
- **Disjunction** ($P \vee Q$): P or Q is true if either P, Q, or both are true.
- **Implication** ($P \to Q$): P implies Q is false only when P is true and Q is false.
- **Biconditional** ($P \leftrightarrow Q$): P and Q are true when both P and Q have the same truth value.

0.2 Truth Tables

A **truth table** shows how the truth value of a compound proposition is determined from the truth values of its components. Here is an example of a truth table for $P \to Q$:

P	Q	$P \to Q$
T	T	T
T	F	F
F	T	T
F	F	T

0.3 Tautology, Contradiction, and Contingency

- A **tautology** is a proposition that is always true, no matter the truth values of its components.
- A **contradiction** is a proposition that is always false.
- A **contingency** is a proposition that is true in some cases and false in others.

0.4 Logical Equivalences

Two propositions are said to be **logically equivalent** if they have the same truth values in all possible scenarios. Some important logical equivalences are:

- **Double negation law**: $\neg(\neg P) \equiv P$
- **De Morgan's laws**:
$$\neg(P \land Q) \equiv \neg P \lor \neg Q$$
$$\neg(P \lor Q) \equiv \neg P \land \neg Q$$
- **Implication law**: $P \to Q \equiv \neg P \lor Q$

1 Predicate Logic (First-Order Logic)

While propositional logic deals with entire statements as a whole, **predicate logic** (also known as first-order logic) deals with the internal structure of propositions. In predicate logic, we use **quantifiers** and **predicates** to express relationships.

1.1 Predicates

A **predicate** is a statement involving variables. It becomes a proposition when the variables are replaced with specific values. For example, let $P(x)$ represent the statement "x is greater than 3." Here, $P(5)$ is a true statement, but $P(2)$ is false.

1.2 Quantifiers

Quantifiers allow us to express propositions involving all or some elements in a domain.

- **Universal quantifier** ($\forall$): $\forall x P(x)$ means that $P(x)$ is true for all values of x.
- **Existential quantifier** ($\exists$): $\exists x P(x)$ means that there is at least one value of x such that $P(x)$ is true.

1.3 Logical Formulas in Predicate Logic

Predicates combined with quantifiers form logical formulas. Here are some examples:

- $\forall x(x > 0 \to x^2 > 0)$: For all x, if x is positive, then x^2 is positive.
- $\exists x(x^2 = 1)$: There exists some x such that $x^2 = 1$.

2 Logical Inference in Predicate Logic

2.1 Rules of Inference

In both propositional and predicate logic, **rules of inference** allow us to derive conclusions from premises. Common rules include:

- **Modus Ponens**: If $P \to Q$ and P are true, then Q is true.
- **Modus Tollens**: If $P \to Q$ and $\neg Q$ are true, then $\neg P$ is true.
- **Universal Instantiation**: From $\forall x P(x)$, we can conclude $P(a)$ for any specific value a.
- **Existential Instantiation**: From $\exists x P(x)$, we can conclude $P(a)$ for some specific a.

3 Worked Example: Translating English Sentences into Predicate Logic

Problem: Translate the following sentence into predicate logic: "All humans are mortal."

Solution: Let $H(x)$ represent "x is a human," and $M(x)$ represent "x is mortal." The logical expression becomes:

$$\forall x (H(x) \to M(x))$$

Problem: Translate "Some cats are black" into predicate logic.

Solution: Let $C(x)$ represent "x is a cat," and $B(x)$ represent "x is black." The logical expression is:

$$\exists x (C(x) \land B(x))$$

Conclusion

Propositional and predicate logic are the foundations of formal reasoning in mathematics and computer science. Mastery of these topics allows for the formalization and analysis of logical statements, making it easier to solve complex problems systematically.

References

- MIT OpenCourseWare: Principles of Applied Mathematics
- MIT OpenCourseWare: Mathematics for Computer Science