2024

FUNDAMENTALS OF MATHEMATICAL LOGIC

Precursor for	*LTN - Smoking Friends Cancer*
Author:	Estevan Hernandez
Revision #:	0.5
Date:	April 18, 2024

Executive Summary

This document serves as an executive summary for three fundamental topics in mathematical logic that provide a strong foundation for understanding Statistical Relational Learning (SRL) and Logical Tensor Networks (LTN). The concepts outlined here are crucial for grasping the SRL LTN lab that follows this lecture. The topics are:

1. **Logical Connectives**:
   - Logical connectives form the building blocks of logical expressions, enabling complex statements and relations to be formulated.
   - The primary connectives include:
     - **And (∧)**: Conjoins two statements, returning true only if both statements are true.
     - **Or (∨)**: Disjoins two statements, returning true if at least one statement is true.
     - **Not (¬)**: Negates a statement, reversing its truth value.
     - **Implies (→)**: Represents a conditional relationship, stating that if the first statement is true, the second must also be true.
   - These connectives allow for the construction of intricate logical rules that can be applied in various domains.
2. **Quantifiers**:
   - Quantifiers specify the quantity of elements that satisfy a certain predicate.
   - The main quantifiers are:
     - **For all (∀)**: Indicates that a predicate holds for every element in a given domain.
     - **Exists (∃)**: Indicates that there is at least one element in a given domain for which the predicate holds true.
   - Quantifiers enable the formulation of general statements about sets of elements, which is critical for representing knowledge in SRL.
3. **Probabilistic Interpretations**:
   - Probabilistic interpretations of logical statements allow for the handling of uncertainty in logical systems.
   - Instead of treating statements as strictly true or false, probabilistic logic assigns a probability to the truth of a statement, allowing for a range of values between 0 and 1.
   - This approach is essential for dealing with the complexities of real-world data and incomplete or contradictory knowledge.

Together, these topics lay the groundwork for the SRL LTN lab, where students will apply the principles of mathematical logic to create and train models using LTNs. By understanding the use of logical connectives, quantifiers, and probabilistic interpretations, students will be equipped to tackle the challenges of representing and reasoning with relational data in complex systems.

Contents

1 Propositional Logic

Propositional logic, also known as sentential logic, propositional calculus, or sentential calculus, is a branch of mathematical logic that deals with propositions and their relationships using logical connectives. A proposition is a statement that can either be true or false, such as "The sky is blue" or "5 is an odd number."

1.1 Key Concepts

1.2 Propositions

These are statements that can be either true or false. In propositional logic, propositions are often represented by uppercase letters, such as A, B, or C.

1.3 Truth Values

Propositions have a truth value, either true or false. In binary terms, true is represented by 1, and false is represented by 0.

1. Connectives: Propositional logic uses logical connectives to combine propositions and express complex logical relationships. The main connectives are:
   - Negation (¬): The negation of a proposition is true if the original proposition is false, and false if the original proposition is true. For example, ¬A is the negation of A.
   - Conjunction (∧): The conjunction of two propositions is true only if both propositions are true. For example, A ∧ B is true only if both A **and** B are true.
   - Disjunction (∨): The disjunction of two propositions is true if at least one of the propositions is true. For example, A ∨ B is true if either A **or** B is true.
   - Conditional (→): The conditional, also known as implication, is true unless a true proposition implies a false one. For example, A → B is false only if A is true and B is false.
   - Biconditional (↔): The biconditional, also known as equivalence, is true if both propositions have the same truth value. For example, A ↔ B is true if **both** A **and** B are either true or false.

1.4 Truth Tables

Truth tables are used to visualize the truth values of propositions and their combinations using connectives. Each row of a truth table represents a possible combination of truth values for the propositions, while the columns represent the truth values of the propositions and the resulting compound propositions.

1.5 Tautology

A proposition that is always true, regardless of the truth values of its constituent propositions. For example, A ∨ ¬A is a tautology because either A is true or ¬A is true in all cases.

1.6 Contradiction

A proposition that is always false, regardless of the truth values of its constituent propositions. For example, $A \land \neg A$ is a contradiction because A and ¬A cannot both be true at the same time.

1.7 Contingent

A proposition that is neither a tautology nor a contradiction; its truth value depends on the truth values of its constituent propositions.

1.8 Applications

Propositional logic is widely used in various fields, such as computer science, electrical engineering, and philosophy. Some common applications include:

- Workflow Problems: Designing and optimizing processes or workflows using logical relationships and constraints.
- Computer Logic Gates: Building and understanding digital circuits using logical operations.
- Game Strategies: Developing and analyzing strategies for games that involve decision-making based on logical reasoning.

2 Predicate Logic

2.1 Why is Propositional Logic Not Enough?

Propositional logic is limited in its ability to express relationships between individuals or entities. It works with atomic propositions that can be either true or false but lacks the ability to describe statements involving quantifiers (e.g., "all," "some") or relationships (e.g., "is taller than").

For example, consider the argument:

- All men are mortal.
- Aristotle is a man.
- Therefore, Aristotle is mortal.

In propositional logic, there is no connection between the statements (A, B, C), so the argument cannot be formalized effectively. Predicate logic addresses this limitation by introducing predicates and quantifiers.

2.2 Predicates

Predicates are functions that return true or false based on their arguments. Predicates express properties or relationships involving entities and can be classified according to their arity (number of arguments). They express properties or relationships involving entities.

- **Monadic Predicates**: Predicates with one argument, such as Tall(x) (x is tall).
- **Dyadic Predicates**: Predicates with two arguments, such as Owes(x, y) (x owes money to y).

- **n-Adic Predicates**: Predicates with n arguments, such as Borrowed(x, y, z) (x borrowed y from z).

Conventionally, predicates are denoted by capital letters (e.g., Tall(x), Owes(x, y)), while variables are denoted by lowercase letters (e.g., x, y, z). Constants, which represent specific objects, are also denoted by lowercase letters (e.g., a, b, c).

2.3 Quantifiers

Quantifiers allow us to express statements about entire sets of objects.

- **Universal Quantifier (∀)**: Applies a predicate to all members of the universe of discourse. For example, ∀x Tall(x) means "Everyone is tall."
- **Existential Quantifier (∃):** Guarantees that the predicate applies to at least one member of the universe of discourse. For example, ∃x CanDance(x) means 'There exists at least one person who can dance."

2.4 Combining Quantifiers

In the given propositions, "L" in "Lxy" represents a binary relation or predicate that relates two elements, x and y. The statement "Lxy" can be interpreted as "x likes y" or "x is less than y," depending on the context of the relation. In the context of quantifiers, the propositions make assertions about how elements (e.g., numbers or people) relate to each other under the defined predicate "L."

Quantifiers can be combined to form more complex propositions:

- **For all, there exists:** ∀x ∃y Lxy means 'For every number x, there exists a number y such that y is less than x.
- **Existential quantifier followed by universal quantifier**: ∃x ∀y Lyx means "There is somebody who everyone likes."

Quantifiers can be combined in series, such as ∃x∃y or ∀x∀y, to express multiple quantifications.

2.5 Formal Definitions

Formal definitions outline the structure of predicate logic:

- **Symbols**: Include predicates, constants, variables, logical connectives, parentheses, and quantifiers.
- **Expressions**: Strings of symbols.
- **Terms**: Constants or variables.
- **Atomic Formula**: Predicate followed by its terms.
- **Well-formed formula (wff)**: An atomic formula, its negation, or combinations using logical connectives.
- **Proposition**: A wff with no free variables.

2.6 Identity

Identity (=) is a two-place predicate indicating equivalence. It satisfies properties of symmetry, transitivity, and reflexivity.

For example, "Liz is the tallest spy" can be formalized using predicate logic with identity:

- UD: All people.
- Sx: x is a spy.
- Txy: x is taller than y.
- l: Liz.

$\forall x\ ((Sx \land \neg(x = l)) \rightarrow Tlx)$ means "For every spy who isn't Liz, Liz is taller than them." [1].

2.7 Definite Descriptions

A definite description identifies a specific object, while indefinite descriptions do not. For example:

- **Definite description**: "The king of France is bald" involves the definite description "the king of France," which may not refer to an existing object.
- **Indefinite description**: "Some dog is annoying" uses an indefinite description "some dog."

2.8 Proofs in Predicate Logic

Proofs involve transforming propositions using inference rules. Key rules include:

- **Universal Elimination**: Replace ∀x A with A by substituting a constant for x.
- **Existential Introduction**: Replace A with ∃x A by substituting a variable for a constant.
- **Quantifier Negation**: Negation of universal quantifiers can be transformed to existential quantifiers and vice versa.

These rules and other proof techniques allow for the formal derivation of conclusions from premises.

By understanding predicate logic and its components, you can work with formal reasoning and modeling relationships in various contexts.

3 Fuzzy Logic

Fuzzy logic is a logical and computational framework that deals with reasoning and decision-making in situations characterized by uncertainty and imprecision. It is based on the theory of fuzzy sets, which allow for varying degrees of membership within a set, rather than strict binary membership (true/false). This flexibility enables fuzzy logic systems to handle the ambiguity and complexity of real-world scenarios more effectively than traditional binary logic systems.

3.1 Core Concepts

3.1.1 Fuzzy Sets

In traditional set theory, an element is either a member of a set or it is not (0 or 1). In fuzzy logic, membership is not binary but rather exists on a continuum between 0 and 1. This allows for partial membership in a set. For example, someone can be "sort of tall" or "mostly tall" rather than strictly tall or short.

3.1.2 Linguistic Variables

Fuzzy logic uses linguistic variables—variables that take on values described in natural language (e.g., 'hot,' 'cold,' 'slow,' 'fast')—to represent concepts and quantities in a way that aligns more closely with human understanding. These linguistic variables are then mapped to numerical values through membership functions.

3.1.3 Fuzzy Rules

Fuzzy logic systems often rely on a set of if-then rules, like those used in traditional rule-based systems, but the antecedents and consequents can be fuzzy. For example, "If the weather is warm, then the air conditioner should run at medium speed" translates human reasoning into fuzzy rule-based computation.

3.1.4 Membership Functions

These are curves that define how each point in the input space is mapped to a membership value between 0 and 1. These functions represent how fuzzy sets behave and vary with different degrees of membership.

3.1.5 Implication and Inference

The process of deriving conclusions from a set of fuzzy rules involves evaluating the degree of truth in the antecedent and applying it to the consequent. The outcome is often a fuzzy set, which can be further processed.

3.1.6 Defuzzification

After processing input through fuzzy rules and reaching an output fuzzy set, the system often needs to convert this set into a single output value for decision-making or control purposes. Defuzzification methods such as centroid, mean of maxima, and others can be used for this purpose.

3.2 Applications

Fuzzy logic has found applications across a wide range of fields and industries, including:

- **Consumer Electronics**: In devices like washing machines, cameras, and microwave ovens to enhance usability and performance.
- **Industrial Control**: In process control systems to manage complex industrial processes.
- **Medical Instrumentation**: For monitoring and diagnostics, allowing for interpretation of imprecise data.

- **Decision-Support Systems**: In aiding human decision-making in complex scenarios.
- **Financial Portfolio Management**: To handle the uncertainty in market behavior and optimize investment strategies.
- **Neuro-Fuzzy Systems**: Combining fuzzy logic with neural networks to create adaptive systems capable of learning and improving over time.

3.3 Advantages

- **Intuitive and Easy to Understand**: Fuzzy logic systems align closely with human reasoning and intuition.
- **Flexible and Adaptable**: These systems can be easily modified to incorporate new rules or adjust existing ones.
- **Tolerant of Imprecise Data**: Fuzzy logic thrives in environments where data is uncertain or incomplete.
- **Suitable for Nonlinear Functions**: Can model complex systems with nonlinear behaviors effectively.
- **Utilizes Expert Knowledge**: Can incorporate insights from subject matter experts directly into the system.

3.4 Limitations and When Not to Use

While fuzzy logic is a powerful tool, it may not always be the most efficient choice. For example, in scenarios where precise, deterministic decision-making is required, or when dealing with problems that can be accurately described using linear models, traditional binary logic or other precise computational methods may be more appropriate.

In summary, fuzzy logic provides a robust and flexible approach to modeling and solving problems characterized by ambiguity and complexity. By approximating human reasoning and bridging the gap between precise and imprecise, it has established itself as a key player in the field of artificial intelligence and control systems.

4 Conclusion

As students' progress to the lab, they will benefit from the ability to construct, evaluate, and optimize models using the principles of mathematical logic. This will prepare them for advanced research and professional work in fields such as artificial intelligence, machine learning, and data science, where the ability to handle complex logical and probabilistic relationships is essential.

Lab Assignment: Logic Tensor Network (LTN) with Smokes-Friends-Cancer Example

5 References

[1] E. B. Agnishom Chattopadhyay, "Predicate Logic," [Online]. Available: https://brilliant.org/wiki/predicate-logic/.

[2] "Logic Tensor Networks," [Online]. Available: https://github.com/logictensornetworks/logictensornetworks.

[3] S. Davies, "Libretexts," [Online]. Available: https://math.libretexts.org/Bookshelves/Combinatorics_and_Discrete_Mathematics/A_Cool_Brisk_Walk_Through_Discrete_Mathematics_(Davies)/08%3A_Logic/8.2%3A_Predicate_logic.

[4] MathWorks, "What Is Fuzzy Logic?," [Online]. Available: https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html.

[5] MathWorks, "Foundations of Fuzzy Logic," [Online]. Available: https://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html.

[6] Brilliant.org, "Propositional Logic," [Online]. Available: https://brilliant.org/wiki/propositional-logic/.