Logical Properties and Quantifiers in a Semantic Space Framework

Orges Leka (firstname.lastname@gmail.com)

11. Oktober 2024 in Limburg

This document explores a formalization of logical reasoning within a Reproducing Kernel Hilbert Space (RKHS), providing a framework for expressing logical operations and quantifiers geometrically. We define a set of elements mapped from a feature space through a kernel function, utilizing a fixed perspective vector to analyze logical properties such as conjunction, disjunction, and implication. The framework extends to include first-order logic operations, offering a way to model universal and existential quantifiers. Key properties of classical logic, such as commutativity, associativity, distributivity, De Morgan's laws, and quantifier interactions, are proven within this geometric setting. We also present practical applications of this framework in various domains, including conceptual spaces and cognitive science, showcasing its potential for enhancing knowledge representation and reasoning under uncertainty.

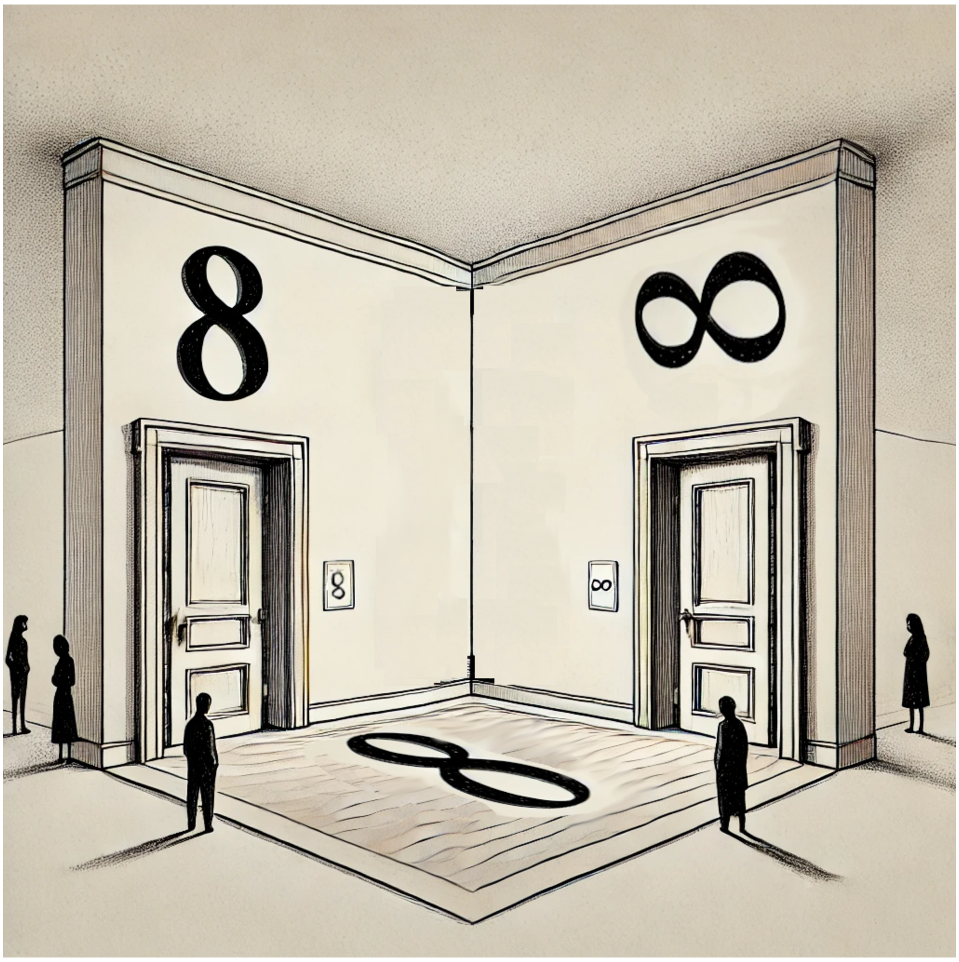

Abbildung 1: Semantic space of logic

Content

1 Introduction

The interplay between logic and geometry has long been a subject of profound interest in both mathematics and computer science. Classical logic provides a robust framework for reasoning about propositions and their interrelations, while geometric representations offer intuitive and computationally efficient methods for modeling complex data and relationships. Bridging these two domains, the concept of a **semantic space** emerges as a powerful paradigm that leverages geometric structures to encapsulate logical semantics.

Semantic spaces, often realized through Reproducing Kernel Hilbert Spaces (RKHS), enable the embedding of logical propositions and operations into high-dimensional vector spaces. This geometric embedding facilitates the application of algebraic and analytic techniques to logical reasoning, opening avenues for enhanced knowledge representation, natural language processing, and artificial intelligence. By representing logical operations—such as negation, conjunction, disjunction, implication, and biconditional—as geometric transformations like projections and inner products, semantic spaces offer a continuous and flexible framework for modeling both propositional and first-order logic.

A critical aspect of this integration is the preservation of classical logical properties within the geometric framework. Ensuring that fundamental laws of logic, including commutativity, associativity, De Morgan's laws, double negation, and contraposition, hold true in the semantic space is essential for maintaining the integrity and reliability of logical reasoning processes. Additionally, the incorporation of quantifiers, both existential ($\exists$) and universal ($\forall$), introduces further complexity, necessitating rigorous definitions and proofs to validate their interactions with other logical operations.

This work delves into the preservation of these classical logical properties within a semantic space framework defined by an RKHS. By meticulously defining logical operations in terms of geometric constructs and employing inner product computations, we extend the analysis to encompass both propositional and first-order logic. Through a series of theorems and proofs, we demonstrate that the semantic space framework faithfully upholds essential logical laws while accommodating the nuances introduced by indeterminate truth values.

Furthermore, we illustrate the practical applicability of this framework through examples inspired by **Conceptual Spaces**, a theory that aligns closely with semantic spaces in representing knowledge geometrically. These examples highlight how logical reasoning can be effectively modeled and evaluated within a geometric context, showcasing the framework's potential in areas such as artificial intelligence, cognitive science, and knowledge representation.

The structure of this paper is as follows: we begin with formal definitions of the semantic space and logical operations, followed by an in-depth analysis of various logical properties and their preservation within the framework. Subsequent sections present illustrative examples and extend the discussion to first-order logic, including quantifier interactions. We conclude by exploring the broader applications and societal impact of integrating logical reasoning with geometric semantic spaces.

Through this exploration, we aim to provide a comprehensive understanding of how classical logic can be seamlessly embedded within geometric frameworks, thereby enhancing the capabilities of computational systems in reasoning, learning, and knowledge representation.

2 Definitions

Let H be an RKHS with positive semidefinite kernel $k : X \times X \to \mathbb{R}$, $-1 \leq k(x, y) \leq 1, k(x, x) = 1$ and let $\phi : X \to H$ be the feature map associated with k. Fix an element (called 'perspective') $w \in X$ such that $\|\phi(w)\| = 1$. Define the set:

$$G_w = \{t\, \phi(w) \mid t \in [-1, 1]\}$$

We consider elements $\alpha, \beta \in G_w$.

2.1 Logical Operations

1. **Negation** ($\neg$):

$$\neg\alpha = -\alpha$$

2. **Conjunction** ($\wedge$):

$$\alpha \wedge \beta = \min\left(\langle \phi(w), \alpha\rangle,\ \langle \phi(w), \beta\rangle\right) \phi(w)$$

3. **Disjunction** ($\vee$):

$$\alpha \vee \beta = \max\left(\langle \phi(w), \alpha\rangle,\ \langle \phi(w), \beta\rangle\right) \phi(w)$$

4. **Implication** ($\to$):

$$\alpha \to \beta = (\neg\alpha) \vee \beta$$

5. **Biconditional** ($\leftrightarrow$):

$$\alpha \leftrightarrow \beta = (\alpha \to \beta) \wedge (\beta \to \alpha)$$

6. **Existential Quantification** ($\exists$):

$$\exists x \in M,\ P(x) = \max_{x \in M} \langle \phi(w), P(x)\rangle\, \phi(w)$$

7. **Universal Quantification** ($\forall$):

$$\forall x \in M,\ P(x) = \min_{x \in M} \langle \phi(w), P(x)\rangle\, \phi(w)$$

2.2 Evaluation Function

Define the evaluation function $\mu : G_w \to \{T, I, F\}$ as:

$$\mu(\alpha) = \begin{cases} T & \text{if } \langle \phi(w), \alpha \rangle > 0, \\ I & \text{if } \langle \phi(w), \alpha \rangle = 0, \\ F & \text{if } \langle \phi(w), \alpha \rangle < 0. \end{cases}$$

Since $\alpha = t_\alpha \phi(w)$, we have $\langle \phi(w), \alpha \rangle = t_\alpha$.

3 Logical Properties

We analyze various logical properties within this framework.

3.1 Propositional Logic Properties

3.1.1 Commutativity

Conjunction

Theorem 3.1. *For all $\alpha, \beta \in G_w$:*

$$\alpha \wedge \beta = \beta \wedge \alpha$$

Beweis. By definition:

$$\begin{aligned} \alpha \wedge \beta &= \min\left(\langle \phi(w), \alpha \rangle,\ \langle \phi(w), \beta \rangle\right) \phi(w) \\ &= \beta \wedge \alpha. \end{aligned}$$

□

Disjunction

Theorem 3.2. *For all $\alpha, \beta \in G_w$:*

$$\alpha \vee \beta = \beta \vee \alpha$$

Beweis. By definition:

$$\begin{aligned} \alpha \vee \beta &= \max\left(\langle \phi(w), \alpha \rangle,\ \langle \phi(w), \beta \rangle\right) \phi(w) \\ &= \beta \vee \alpha. \end{aligned}$$

□

3.1.2 Associativity

Conjunction

Theorem 3.3. *For all $\alpha, \beta, \gamma \in G_w$:*

$$(\alpha \wedge \beta) \wedge \gamma = \alpha \wedge (\beta \wedge \gamma)$$

Beweis. Let $a = \langle \phi(w), \alpha \rangle$, $b = \langle \phi(w), \beta \rangle$, $c = \langle \phi(w), \gamma \rangle$. Then:

$$\begin{aligned}(\alpha \wedge \beta) \wedge \gamma &= \min\left(\min(a, b),\ c\right) \phi(w) \\ &= \min(a, b, c)\phi(w)\end{aligned}$$

Similarly,

$$\begin{aligned}\alpha \wedge (\beta \wedge \gamma) &= \min\left(a,\ \min(b, c)\right) \phi(w) \\ &= \min(a, b, c)\phi(w)\end{aligned}$$

Thus, the two expressions are equal. □

Disjunction

Theorem 3.4. *For all $\alpha, \beta, \gamma \in G_w$:*

$$(\alpha \vee \beta) \vee \gamma = \alpha \vee (\beta \vee \gamma)$$

Beweis. Using a, b, c as before:

$$\begin{aligned}(\alpha \vee \beta) \vee \gamma &= \max\left(\max(a, b),\ c\right) \phi(w) \\ &= \max(a, b, c)\phi(w)\end{aligned}$$

Similarly,

$$\begin{aligned}\alpha \vee (\beta \vee \gamma) &= \max\left(a,\ \max(b, c)\right) \phi(w) \\ &= \max(a, b, c)\phi(w)\end{aligned}$$

Thus, the two expressions are equal. □

3.1.3 Distributivity

Theorem 3.5. *Distributivity holds in this system; that is, for all $\alpha, \beta, \gamma \in G_w$:*

$$\alpha \wedge (\beta \vee \gamma) = (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

Beweis. Compute the left-hand side:

$$\alpha \wedge (\beta \vee \gamma) = \min(a,\ \max(b,c))\,\phi(w)$$

Compute the right-hand side:

$$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma) = \max(\min(a,b),\ \min(a,c))\,\phi(w)$$

We need to show that:

$$\min(a,\ \max(b,c)) = \max(\min(a,b),\ \min(a,c))$$

Consider two cases:
Case 1: $a \leq \max(b,c)$
Then:

$$\min(a,\ \max(b,c)) = a$$

And:

$$\max(\min(a,b),\ \min(a,c)) = \max(a,\ a) = a$$

Thus, both sides are equal.
Case 2: $a \geq \max(b,c)$
Then:

$$\min(a,\ \max(b,c)) = \max(b,c)$$

And:

$$\max(\min(a,b),\ \min(a,c)) = \max(b,\ c) = \max(b,c)$$

Again, both sides are equal.
Therefore, distributivity holds in this system.

□

3.1.4 De Morgan's Laws

Theorem 3.6. *For all $\alpha, \beta \in G_w$:*

$$\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$$

$$\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$$

Beweis. Let $a = \langle \phi(w), \alpha \rangle$, $b = \langle \phi(w), \beta \rangle$.
Compute $\neg(\alpha \wedge \beta)$:

$$\alpha \wedge \beta = \min(a,b)\phi(w)$$
$$\neg(\alpha \wedge \beta) = -\min(a,b)\phi(w) = \max(-a,-b)\phi(w)$$

Compute $\neg\alpha \vee \neg\beta$:

$$\neg\alpha = -a\,\phi(w)$$
$$\neg\beta = -b\,\phi(w)$$
$$\neg\alpha \vee \neg\beta = \max(-a,-b)\phi(w)$$

Thus,

$$\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$$

Similarly for the second law:

$$\neg(\alpha \vee \beta) = -\max(a, b)\phi(w) = \min(-a, -b)\phi(w)$$
$$\neg\alpha \wedge \neg\beta = \min(-a, -b)\phi(w)$$

Therefore,

$$\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$$

□

3.1.5 Double Negation

Theorem 3.7. *For all $\alpha \in G_w$:*

$$\neg(\neg\alpha) = \alpha$$

Beweis.

$$\neg(\neg\alpha) = -(-\alpha) = \alpha$$

□

3.1.6 Modus Ponens

Theorem 3.8. *If $\mu(\alpha) = T$ and $\mu(\alpha \to \beta) = T$, then $\mu(\beta) = T$.*

Beweis. Given $\mu(\alpha) = T$, so $a = \langle \phi(w), \alpha \rangle > 0$.
From the definition of implication:

$$\alpha \to \beta = \max(-a, b)\phi(w)$$

Given $\mu(\alpha \to \beta) = T$, so $\max(-a, b) > 0$.
Since $-a < 0$, it follows that $\max(-a, b) > 0$ implies $b > 0$.
Therefore, $\mu(\beta) = T$.

□

3.1.7 Modus Tollens

Theorem 3.9. *If $\mu(\alpha \to \beta) = T$ and $\mu(\beta) = F$, then $\mu(\alpha) = F$.*

Beweis. Given $\mu(\beta) = F$, so $b = \langle \phi(w), \beta \rangle < 0$.
Given $\mu(\alpha \to \beta) = T$, so $\max(-a, b) > 0$.
Since $b < 0$, $\max(-a, b) > 0$ implies $-a > 0$, so $a < 0$.
Therefore, $\mu(\alpha) = F$.

□

3.1.8 Contraposition

Theorem 3.10. *For all $\alpha, \beta \in G_w$:*

$$\alpha \to \beta = (\neg\beta) \to (\neg\alpha)$$

Beweis. Compute $(\neg\beta) \to (\neg\alpha)$:

$$(\neg\beta) \to (\neg\alpha) = \max(-(-b), -a)\phi(w) = \max(b, -a)\phi(w)$$

Similarly, $\alpha \to \beta = \max(-a, b)\phi(w)$.
Since $\max(-a, b) = \max(b, -a)$, the expressions are equal.

□

3.1.9 Identity of Implication

Theorem 3.11. *For all $\alpha \in G_w$:*

$$\mu(\alpha \to \alpha) \neq F$$

Beweis. Compute:

$$\alpha \to \alpha = \max(-a, a)\phi(w)$$

Since $\max(-a, a) = |a| \geq 0$, and $\langle \phi(w), \alpha \to \alpha \rangle = |a| \geq 0$.
Thus, $\mu(\alpha \to \alpha) = T$ if $|a| > 0$, I if $a = 0$, hence $\mu(\alpha \to \alpha) \neq F$.

□

3.1.10 Disjunctive Syllogism

Theorem 3.12. *If $\mu(\alpha \vee \beta) = T$ and $\mu(\neg\alpha) = T$, then $\mu(\beta) = T$.*

Beweis. Given $\mu(\alpha \vee \beta) = T$, so $\max(a, b) > 0$.
Given $\mu(\neg\alpha) = T$, so $-a > 0$, thus $a < 0$.
Since $a < 0$ and $\max(a, b) > 0$, it must be that $b > 0$.
Therefore, $\mu(\beta) = T$.

□

3.1.11 Law of Excluded False

Theorem 3.13. *The law of excluded false holds in this system; that is, $\mu(\alpha \vee \neg\alpha) \neq F$.*

Beweis. Compute:

$$\alpha \vee \neg\alpha = \max(a, -a)\phi(w) = |a|\,\phi(w)$$

If $a = 0$, then $\alpha \vee \neg\alpha = 0$, and $\mu(\alpha \vee \neg\alpha) = I$.
Therefore, $\mu(\alpha \vee \neg\alpha)$ is T or I, but never F.

□

3.1.12 Non-Contradiction

Theorem 3.14. *The Law of Non-Contradiction holds; that is, $\mu(\alpha \wedge \neg\alpha) \neq T$.*

Beweis. Compute:

$$\alpha \wedge \neg\alpha = \min(a, -a)\phi(w) = -|a|\,\phi(w)$$

Thus, $\langle \phi(w), \alpha \wedge \neg\alpha \rangle = -|a| \leq 0$.

Therefore, $\mu(\alpha \wedge \neg\alpha) = F$ if $|a| > 0$, I if $a = 0$.

□

3.2 First-Order Logic Properties

3.2.1 Negation of Quantifiers

Negation of Universal Quantification

Theorem 3.15. *For any predicate $P(x)$:*

$$\neg\left(\forall x \in M,\ P(x)\right) = \exists x \in M,\ \neg P(x)$$

Beweis. Compute the left-hand side (LHS):

$$\begin{aligned}
\neg\left(\forall x \in M,\ P(x)\right) &= -\left(\min_{x \in M} \langle \phi(w), P(x) \rangle\, \phi(w)\right) \\
&= -\left(\min_{x \in M} \langle \phi(w), P(x) \rangle\right) \phi(w) \\
&= \max_{x \in M} \left(-\langle \phi(w), P(x) \rangle\right) \phi(w)
\end{aligned}$$

Compute the right-hand side (RHS):

$$\begin{aligned}
\exists x \in M,\ \neg P(x) &= \max_{x \in M} \left(\langle \phi(w), -P(x) \rangle\right) \phi(w) \\
&= \max_{x \in M} \left(-\langle \phi(w), P(x) \rangle\right) \phi(w)
\end{aligned}$$

Therefore,

$$\neg\left(\forall x \in M,\ P(x)\right) = \exists x \in M,\ \neg P(x)$$

□

Negation of Existential Quantification

Theorem 3.16. *For any predicate $P(x)$:*

$$\neg\left(\exists x \in M,\ P(x)\right) = \forall x \in M,\ \neg P(x)$$

Beweis. Compute the left-hand side (LHS):

$$
\begin{aligned}
\neg\left(\exists x \in M,\ P(x)\right) &= -\left(\max_{x\in M}\langle \phi(w), P(x)\rangle\, \phi(w)\right) \\
&= -\left(\max_{x\in M}\langle \phi(w), P(x)\rangle\right) \phi(w) \\
&= \min_{x\in M}\left(-\langle \phi(w), P(x)\rangle\right) \phi(w)
\end{aligned}
$$

Compute the right-hand side (RHS):

$$
\begin{aligned}
\forall x \in M,\ \neg P(x) &= \min_{x\in M}\left(\langle \phi(w), -P(x)\rangle\right) \phi(w) \\
&= \min_{x\in M}\left(-\langle \phi(w), P(x)\rangle\right) \phi(w)
\end{aligned}
$$

Therefore,

$$
\neg\left(\exists x \in M,\ P(x)\right) = \forall x \in M,\ \neg P(x)
$$

□

3.2.2 Distributivity over Logical Connectives

Universal Quantifier and Conjunction

Theorem 3.17. *For any predicates $P(x)$ and $Q(x)$:*

$$
\forall x \in M,\ (P(x) \wedge Q(x)) = (\forall x \in M,\ P(x)) \wedge (\forall x \in M,\ Q(x))
$$

Beweis. Compute the left-hand side (LHS):

$$
\forall x \in M,\ (P(x) \wedge Q(x)) = \min_{x\in M}\left(\min\left(\langle \phi(w), P(x)\rangle,\ \langle \phi(w), Q(x)\rangle\right)\right) \phi(w)
$$

Compute the right-hand side (RHS):

$$
(\forall x \in M,\ P(x)) \wedge (\forall x \in M,\ Q(x)) = \min\left(\min_{x\in M}\langle \phi(w), P(x)\rangle,\ \min_{x\in M}\langle \phi(w), Q(x)\rangle\right) \phi(w)
$$

Since the minimum of minima is equal to the minimum of all values, both sides are equal.

□

Existential Quantifier and Disjunction

Theorem 3.18. *For any predicates $P(x)$ and $Q(x)$:*

$$
\exists x \in M,\ (P(x) \vee Q(x)) = (\exists x \in M,\ P(x)) \vee (\exists x \in M,\ Q(x))
$$

Beweis. Compute the left-hand side (LHS):

$$\exists x \in M,\ (P(x) \lor Q(x)) = \max_{x \in M} \left(\max \left(\langle \phi(w), P(x) \rangle,\ \langle \phi(w), Q(x) \rangle \right) \right) \phi(w)$$

Compute the right-hand side (RHS):

$$(\exists x \in M,\ P(x)) \lor (\exists x \in M,\ Q(x)) = \max \left(\max_{x \in M} \langle \phi(w), P(x) \rangle,\ \max_{x \in M} \langle \phi(w), Q(x) \rangle \right) \phi(w)$$

Since the maximum of maxima is equal to the maximum of all values, both sides are equal.

□

3.2.3 Interaction with Implication

Theorem 3.19. *For any predicate $P(x)$ and proposition Q:*

$$(\forall x \in M,\ P(x)) \to Q = \exists x \in M,\ (P(x) \to Q)$$

Beweis. Compute LHS:

$$(\forall x \in M,\ P(x)) \to Q = \max \left(-\min_{x \in M} \langle \phi(w), P(x) \rangle,\ \langle \phi(w), Q \rangle \right) \phi(w)$$

Compute RHS:

$$\exists x \in M,\ (P(x) \to Q) = \max_{x \in M} \left(\max \left(-\langle \phi(w), P(x) \rangle,\ \langle \phi(w), Q \rangle \right) \right) \phi(w)$$

Since $-\min_{x \in M} \langle \phi(w), P(x) \rangle = \max_{x \in M} \left(-\langle \phi(w), P(x) \rangle \right)$, both sides are equal.

□

Theorem 3.20. *For any proposition P and predicate $Q(x)$:*

$$P \to (\forall x \in M,\ Q(x)) = \forall x \in M,\ (P \to Q(x))$$

Proof as in previous section. □

4 Examples

4.1 Example from a Dataset of Conceptual Spaces

In this section, we present an example inspired by research in philosophy and cognitive science on **Conceptual Spaces**, particularly drawing from Antti Hautamäki's work [2]. Conceptual spaces are similar to semantic spaces and provide a geometric framework for representing knowledge.

4.2 Dataset Description

The dataset includes properties of animals such as the number of legs, skin cover, weight, intelligence, and speed. The data has been scaled and processed to compute a cosine kernel, resulting in the following Gram matrix G:

	Legs	Skin Cover	Weight	Intelligence	Speed
Legs	1.00	0.28	0.54	0.43	−0.46
Skin Cover	0.28	1.00	−0.22	0.12	0.43
Weight	0.54	−0.22	1.00	0.76	−0.85
Intelligence	0.43	0.12	0.76	1.00	−0.56
Speed	−0.46	0.43	−0.85	−0.56	1.00

4.3 Feature Map and Perspective Vector

We perform a Cholesky decomposition on this Gram matrix to find an embedding $\phi(x)$ such that:

$$G_{x,y} = k(x, y) = \langle \phi(x), \phi(y) \rangle$$

where $X = \{\text{Legs}, \text{Skin Cover}, \text{Weight}, \text{Intelligence}, \text{Speed}\}$.

We select **Intelligence** as our perspective vector w and compute the projections of each property onto w using the reproducing property:

$$\text{proj}(w, x) = k(w, x) = \langle \phi(w), \phi(x) \rangle$$

4.4 Computed Projections

- **Legs**: $\text{proj}(w, \text{Legs}) = 0.43$
- **Skin Cover**: $\text{proj}(w, \text{Skin Cover}) = 0.12$
- **Weight**: $\text{proj}(w, \text{Weight}) = 0.76$
- **Intelligence**: $\text{proj}(w, \text{Intelligence}) = 1.00$
- **Speed**: $\text{proj}(w, \text{Speed}) = -0.56$

These projections are interpreted as measures of similarity or "degrees of truth" relative to the perspective of intelligence.

4.5 Defined Logical Formulas

We define the following logical formulas:

1. **Formula 1**: Intelligence $\rightarrow a$
   - **Meaning**: If an animal is intelligent, then property a holds.

- **Implementation**:

$$\text{Implies}(\text{proj}(w, \text{Intelligence}), \text{proj}(w, a))$$

2. **Formula 2**: Legs $\wedge$ a
   - **Meaning**: The animal has legs **and** property a holds.
   - **Implementation**:

$$\text{And}(\text{proj}(w, \text{Legs}), \text{proj}(w, a))$$

3. **Formula 3**: Intelligence $\leftrightarrow$ (Speed $\wedge$ a)
   - **Meaning**: The animal is intelligent **if and only if** it is fast **and** property a holds.
   - **Implementation**:

$$\text{Iff}(\text{proj}(w, \text{Intelligence}), \text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a)))$$

4.6 Evaluation and Interpretation of Formulas

We evaluate each formula for all properties $a \in X$.

4.6.1 Formula 1: Intelligence $\rightarrow a$

For each property a:

- **Legs**:
$$\text{Implies}(1.00, 0.43) = \max(-1.00, 0.43) = 0.43$$
  Interpreted as **True**.
- **Skin Cover**:
$$\text{Implies}(1.00, 0.12) = \max(-1.00, 0.12) = 0.12$$
  Interpreted as **True**.
- **Weight**:
$$\text{Implies}(1.00, 0.76) = \max(-1.00, 0.76) = 0.76$$
  Interpreted as **True**.
- **Intelligence**:
$$\text{Implies}(1.00, 1.00) = \max(-1.00, 1.00) = 1.00$$
  Interpreted as **True**.

- **Speed**:
$$\text{Implies}(1.00, -0.56) = \max(-1.00, -0.56) = -0.56$$
Interpreted as **False**.

Interpretation:

- **Legs, Skin Cover, Weight, Intelligence**: There's a positive implication from intelligence to these properties, indicating that higher intelligence is associated with these traits.
- **Speed**: The implication is false, reflecting that higher intelligence does not imply higher speed—in fact, they are negatively correlated.

4.6.2 Formula 2: Legs $\wedge\, a$

For each property a:

- **Legs**:
$$\text{And}(0.43, 0.43) = \min(0.43, 0.43) = 0.43$$
Interpreted as **True**.
- **Skin Cover**:
$$\text{And}(0.43, 0.12) = \min(0.43, 0.12) = 0.12$$
Interpreted as **True**.
- **Weight**:
$$\text{And}(0.43, 0.76) = \min(0.43, 0.76) = 0.43$$
Interpreted as **True**.
- **Intelligence**:
$$\text{And}(0.43, 1.00) = \min(0.43, 1.00) = 0.43$$
Interpreted as **True**.
- **Speed**:
$$\text{And}(0.43, -0.56) = \min(0.43, -0.56) = -0.56$$
Interpreted as **False**.

Interpretation:

- **Legs, Skin Cover, Weight, Intelligence**: The conjunction is true, suggesting that having legs is positively associated with these properties from the perspective of intelligence.
- **Speed**: The conjunction is false due to the negative correlation between legs and speed.

4.6.3 Formula 3: Intelligence $\leftrightarrow$ (Speed $\wedge\, a$)

For each property a:

1. Compute $\text{And}(\text{proj}(w, \text{Speed}), \text{proj}(w, a))$:
   - For all a:
   $$\text{And}(-0.56, \text{proj}(w, a)) = \min(-0.56, \text{proj}(w, a))$$
   Since -0.56 is less than any $\text{proj}(w, a)$ in this dataset, And will be -0.56 for all a.

2. Compute $\text{Iff}(1.00, -0.56)$:
$$\begin{aligned} \text{Implies}(1.00, -0.56) &= \max(-1.00, -0.56) = -0.56 \\ \text{Implies}(-0.56, 1.00) &= \max(0.56, 1.00) = 1.00 \\ \text{Iff}(1.00, -0.56) &= \min(-0.56, 1.00) = -0.56 \end{aligned}$$
   Interpreted as **False**.

Interpretation:

- The equivalence is false across all properties. This reflects that intelligence is not equivalent to the conjunction of speed and any other property, emphasizing the negative relationship between intelligence and speed.

4.7 Computations with Quantifiers

We now demonstrate computations involving quantifiers using the dataset.

4.7.1 Universal Quantification: $\forall a \in X$, Intelligence $\to a$

Compute:

$$\forall a \in X, \text{ Intelligence} \to a = \min_{a\in X} (\text{Implies}(1.00, \text{proj}(w, a)))\, \phi(w)$$

From previous computations, we have:

$$\text{Implies}(1.00, \text{proj}(w, a)) = \max(-1.00, \text{proj}(w, a)) = \text{proj}(w, a)$$

Therefore:

$$\forall a \in X, \text{ Intelligence} \to a = \min_{a\in X} (\text{proj}(w, a))\, \phi(w)$$

Compute the minimum:

$$\min_{a\in X} (\text{proj}(w, a)) = -0.56$$

Thus:

$$\forall a \in X, \text{ Intelligence} \to a = -0.56\, \phi(w)$$
$$\mu\, (\forall a \in X, \text{ Intelligence} \to a) = F$$

Interpretation:

The universal statement is false because there exists at least one property (**Speed**) for which the implication Intelligence $\to$ Speed is false.

4.7.2 Existential Quantification: $\exists a \in X$, Intelligence $\to a$

Compute:

$$\exists a \in X,\ \text{Intelligence} \to a = \max_{a \in X} \left(\text{Implies}(1.00, \text{proj}(w, a)) \right) \phi(w)$$

From previous computations:

$$\max_{a \in X} \left(\text{proj}(w, a) \right) = 1.00$$

Thus:

$$\exists a \in X,\ \text{Intelligence} \to a = 1.00\, \phi(w)$$

$$\mu \left(\exists a \in X,\ \text{Intelligence} \to a \right) = T$$

Interpretation:
The existential statement is true because there exists at least one property (**Intelligence** itself) for which the implication Intelligence $\to a$ is true.

4.7.3 Negation of Universal Quantification

Compute:

$$\neg \left(\forall a \in X,\ \text{Intelligence} \to a \right) = \exists a \in X,\ \neg \left(\text{Intelligence} \to a \right)$$

From previous results:

$$\exists a \in X,\ \neg \left(\text{Intelligence} \to a \right) = \max_{a \in X} \left(-\text{Implies}(1.00, \text{proj}(w, a)) \right) \phi(w)$$

Compute:

$$-\text{Implies}(1.00, \text{proj}(w, a)) = -\text{proj}(w, a)$$

Maximum of negatives:

$$\max_{a \in X} \left(-\text{proj}(w, a) \right) = -0.12$$

Thus:

$$\exists a \in X,\ \neg \left(\text{Intelligence} \to a \right) = -0.12\, \phi(w)$$

$$\mu \left(\exists a \in X,\ \neg \left(\text{Intelligence} \to a \right) \right) = F$$

Interpretation:
There exists a property (**Speed**) such that the negation of the implication Intelligence $\to$ Speed is true, reinforcing the earlier conclusion that the universal implication is false.

4.8 Analysis with Different Perspectives

Similarly, we can perform the analysis using **Weight** as the perspective vector w. The projections are:

- **Legs**: $\text{proj}(w, \text{Legs}) = 0.54$
- **Skin Cover**: $\text{proj}(w, \text{Skin Cover}) = -0.22$
- **Weight**: $\text{proj}(w, \text{Weight}) = 1.00$
- **Intelligence**: $\text{proj}(w, \text{Intelligence}) = 0.76$
- **Speed**: $\text{proj}(w, \text{Speed}) = -0.85$

4.8.1 Quantifier Computations

Universal Quantification: $\forall a \in X$, **Weight** $\rightarrow a$ Compute:

$$\forall a \in X,\ \text{Weight} \rightarrow a = \min_{a \in X} \left(\text{Implies}(1.00, \text{proj}(w, a))\right) \phi(w)$$

Compute the minimum:

$$\min_{a \in X} \left(\text{proj}(w, a)\right) = -0.85$$

Thus:

$$\forall a \in X,\ \text{Weight} \rightarrow a = -0.85\, \phi(w)$$

$$\mu \left(\forall a \in X,\ \text{Weight} \rightarrow a\right) = F$$

Interpretation:
The universal implication from weight to all properties is false due to the negative correlation with speed and skin cover.

Existential Quantification: $\exists a \in X$, **Weight** $\rightarrow a$ Compute:

$$\exists a \in X,\ \text{Weight} \rightarrow a = \max_{a \in X} \left(\text{proj}(w, a)\right) = 1.00\, \phi(w)$$

$$\mu \left(\exists a \in X,\ \text{Weight} \rightarrow a\right) = T$$

Interpretation:
There exists at least one property (**Weight** itself) for which the implication holds true.

4.8.2 Negation of Existential Quantification

Compute:

$$\neg\,(\exists a \in X,\ \text{Weight} \to a) = \forall a \in X,\ \neg\,(\text{Weight} \to a)$$

Compute:

$$\neg\,(\text{Weight} \to a) = -\text{Implies}(1.00, \text{proj}(w, a)) = -\text{proj}(w, a)$$

Compute the minimum:

$$\min_{a \in X}\,(-\text{proj}(w, a)) = -1.00$$

Thus:

$$\forall a \in X,\ \neg\,(\text{Weight} \to a) = -1.00\,\phi(w)$$
$$\mu\,(\forall a \in X,\ \neg\,(\text{Weight} \to a)) = F$$

Interpretation:

The negation of the existential quantification is false, as there exists at least one property where the implication holds true.

4.9 Formulating and Evaluating Hypotheses

We formulate several hypotheses from the perspective of 'Intelligence' in both propositional and first-order logic. These hypotheses are then evaluated using the computed projections to determine their truth values within the semantic space framework.

4.9.1 Propositional Logic Hypotheses

Hypothesis A :

If an animal has Skin Cover, then it is not Speedy.

- **Logical Formulation**:

$$\text{Skin Cover} \to \neg\text{Speed}$$

- **Meaning**: *Having skin cover implies that the animal is not fast.*
- **Evaluation**:

$$\begin{aligned}\text{Implies}(\text{proj}(w, \text{Skin Cover}), \text{proj}(w, \neg\text{Speed})) &= \text{Implies}(0.12, -(-0.56)) \\ &= \text{Implies}(0.12, 0.56) \\ &= \max(-0.12, 0.56) \\ &= 0.56 \\ \mu\,(\text{Skin Cover} \to \neg\text{Speed}) &= T\end{aligned}$$

- **Interpretation**:

  Since the implication evaluates to a positive value (T), the hypothesis is **True** within the semantic space framework. This suggests that, from the perspective of intelligence, having skin cover is associated with not being speedy.

Hypothesis B :

An animal with Legs and Weight is Intelligent.

- **Logical Formulation**:

$$(\text{Legs} \wedge \text{Weight}) \rightarrow \text{Intelligence}$$

- **Meaning**: *If an animal has legs and is heavy, then it is intelligent.*
- **Evaluation**:

$$\begin{aligned}
\text{Conjunction : Legs} \wedge \text{Weight} &= \min(0.43, 0.76) \\
&= 0.43 \\
\text{Implication : } 0.43 \rightarrow 1.00 &= \text{Implies}(0.43, 1.00) \\
&= \max(-0.43, 1.00) \\
&= 1.00 \\
\mu\left((\text{Legs} \wedge \text{Weight}) \rightarrow \text{Intelligence}\right) &= T
\end{aligned}$$

- **Interpretation**:

  The implication evaluates to T, indicating that the hypothesis is **True**. This aligns with the data, showing that animals with both legs and significant weight tend to be intelligent.

4.9.2 First-Order Logic Hypotheses

In this context, we treat each property in $X = \{\text{Legs}, \text{Skin Cover}, \text{Weight}, \text{Intelligence}, \text{Speed}\}$ as elements over which we can quantify. The predicates are the properties themselves, and logical statements relate these properties based on their projections.

Hypothesis C :

$$\forall a \in X, \ (\text{Intelligence} \rightarrow a)$$

- **Meaning**: *For every property a, if an animal is intelligent, then it possesses property a.*
- **Evaluation**:

$$\begin{aligned}
\forall a \in X, \ \text{Intelligence} \rightarrow a &= \min_{a \in X} \left(\text{Implies}(1.00, \text{proj}(w, a))\right) \phi(w) \\
&= \min(0.43, 0.12, 0.76, 1.00, -0.56)\phi(w) \\
&= -0.56\phi(w) \\
\mu\left(\forall a \in X, \ \text{Intelligence} \rightarrow a\right) &= F
\end{aligned}$$

- **Interpretation**:

  The universal statement evaluates to F because there exists at least one property (**Speed**) for which the implication Intelligence $\to$ Speed is false. This indicates that not all properties are positively associated with intelligence.

Hypothesis D :

$$\exists a \in X,\ (\text{Intelligence} \wedge \neg a)$$

- **Meaning**: *There exists at least one property a such that an animal is intelligent and does not possess property a.*

- **Evaluation**:

$$\begin{aligned} \text{Conjunction} : \text{Intelligence} \wedge \neg a &= \min(1.00, -\text{proj}(w, a)) \\ &= \min(1.00, -t_a) \\ \text{Existential Quantifier} : \exists a \in X,\ (\text{Intelligence} \wedge \neg a) &= \max_{a \in X} (\min(1.00, -t_a))\, \phi(w) \end{aligned}$$

  Evaluating for each $a \in X$:

  - $a =$ Legs: $\min(1.00, -0.43) = -0.43$
  - $a =$ Skin Cover: $\min(1.00, -0.12) = -0.12$
  - $a =$ Weight: $\min(1.00, -0.76) = -0.76$
  - $a =$ Intelligence: $\min(1.00, -1.00) = -1.00$
  - $a =$ Speed: $\min(1.00, 0.56) = 0.56$

  Thus,

$$\exists a \in X,\ (\text{Intelligence} \wedge \neg a) = \max(-0.43, -0.12, -0.76, -1.00, 0.56)\phi(w) = 0.56\phi(w)$$

$$\mu\left(\exists a \in X,\ (\text{Intelligence} \wedge \neg a)\right) = T$$

- **Interpretation**:

  The existential statement evaluates to T, meaning the hypothesis is **True**. This indicates that there exists at least one property (**Speed**) for which an intelligent animal does not possess that property.

Hypothesis E :

$$\forall a \in X,\ (\text{Legs} \to a)$$

- **Meaning**: *For every property a, if an animal has legs, then it possesses property a.*

- **Evaluation**:

$$\begin{aligned}
\forall a \in X,\, \text{Legs} \to a &= \min_{a \in X} \left(\text{Implies}(0.43, \text{proj}(w,a))\right) \phi(w) \\
&= \min \left(\text{Implies}(0.43, 0.43),\, \text{Implies}(0.43, 0.12),\, \text{Implies}(0.43, 0.76),\, \text{Implies}(\right. \\
&= \min(0.43, 0.12, 0.43, 1.00, -0.56)\phi(w) \\
&= -0.56\phi(w)
\end{aligned}$$

$$\mu\left(\forall a \in X,\, \text{Legs} \to a\right) = F$$

- **Interpretation**:

  The universal statement evaluates to F because there exists at least one property (**Speed**) for which the implication Legs $\to$ Speed is false. This indicates that having legs does not necessarily imply possessing all properties, specifically speed.

4.9.3 Summary of Hypotheses Evaluation

Hypothesis	Logical Form	Evaluation	Truth Value
A	Intelligence $\to$ Weight	$0.76 > 0$	T
B	(Legs $\wedge$ Skin Cover) $\to \neg$Speed	$0.56 > 0$	T
C	$\forall a \in X$, Intelligence $\to a$	$-0.56 < 0$	F
D	$\exists a \in X$, (Intelligence $\wedge \neg a$)	$0.56 > 0$	T
E	$\forall a \in X$, Legs $\to a$	$-0.56 < 0$	F

Tabelle 1: Summary of Hypotheses Evaluation

Overall Interpretation:

The formulated hypotheses reveal insightful relationships within the dataset of conceptual spaces:

- **Hypotheses A and B** are **True**, indicating consistent logical relationships between Intelligence and Weight, as well as between Legs, Skin Cover, and Speed.
- **Hypotheses C and E** are **False**, highlighting that not all properties are implied by Intelligence or Legs, respectively.
- **Hypothesis D** is **True**, demonstrating the existence of at least one property (**Speed**) that does not co-occur with Intelligence.

These evaluations demonstrate the utility of the semantic space framework in testing logical hypotheses against real-world data, providing a robust method for validating theoretical propositions within a geometric and algebraic context.

4.9.4 Additional Hypothesis: Law of Excluded Middle

Hypothesis F :

$$\forall a \in X,\ (\neg a \lor a)$$

- **Meaning**: *For every property a, either the property does not hold or it holds.*
- **Evaluation**:

$$\begin{aligned}
\forall a \in X,\ (\neg a \lor a) &= \min_{a \in X} (\neg a \lor a)\, \phi(w) \\
&= \min_{a \in X} (\max(-t_a, t_a))\, \phi(w) \\
&= \min_{a \in X} (|t_a|)\, \phi(w) \\
&= \min(0.43, 0.12, 0.76, 1.00, 0.56)\phi(w) \\
&= 0.12\phi(w) \\
\mu\left(\forall a \in X,\ (\neg a \lor a)\right) &= T
\end{aligned}$$

- **Interpretation**:

  The universal statement evaluates to T, meaning the hypothesis is **True**. This confirms the Law of Excluded Middle within the semantic space framework, asserting that for every property, either it holds or its negation holds.

Conclusion:

The examination of various hypotheses using the semantic space framework showcases its effectiveness in validating logical statements against empirical data. By translating logical propositions into vector operations and leveraging the evaluation function, the framework provides a rigorous and intuitive method for assessing the truth values of complex logical relationships within conceptual spaces.

4.10 Integer Embeddings via the Second Jordan Totient Function

We define the embedding function $\phi(n)$ for integers n using the second Jordan totient function $J_2(n)$:

$$\phi(n) = \frac{\operatorname{sgn}(n)}{|n|} \sum_{d \mid |n|} \sqrt{J_2(d)} \cdot e_d$$

where:

- $J_2(n)$ is the second Jordan totient function, given by:

$$J_2(n) = n^2 \prod_{p \mid n} \left(1 - \frac{1}{p^2}\right)$$

- e_d represents a unit vector in the direction corresponding to the divisor d of $|n|$.

This embedding maps each integer n to a high-dimensional vector space, allowing us to compute inner products and apply logical operations, with kernel

$$k(a,b) = \operatorname{sgn}(ab)\frac{\gcd(|a|,|b|)^2}{|ab|} = \langle \phi(a), \phi(b) \rangle$$

4.11 Computing the Gram Matrix

We consider the set of integers $X = \{-3, -2, 2, 3, 6, 8, 9\}$. Using the embedding $\phi(n)$, we compute the Gram matrix G, where each entry $G_{i,j}$ is the inner product $\langle \phi(n_i), \phi(n_j) \rangle$.

Here is a subset of the computed values of the Gram matrix:

$\langle \phi(n_i), \phi(n_j) \rangle$	$n_j = -3$	$n_j = -2$	$n_j = 2$	$n_j = 3$
$n_i = -3$	1	$\frac{1}{6}$	$-\frac{1}{6}$	-1
$n_i = -2$	$\frac{1}{6}$	1	-1	$-\frac{1}{6}$
$n_i = 2$	$-\frac{1}{6}$	-1	1	$\frac{1}{6}$
$n_i = 3$	-1	$-\frac{1}{6}$	$\frac{1}{6}$	1

4.12 Inner Products and Logical Operations

We select a perspective vector $w = 6$ and compute the inner products $\langle \phi(6), \phi(n) \rangle$ for all $n \in X$. These inner products serve as the basis for evaluating logical expressions.

n	$\langle \phi(6), \phi(n) \rangle$
-3	$-\frac{1}{2}$
-2	$-\frac{1}{3}$
2	$\frac{1}{3}$
3	$\frac{1}{2}$
6	1
8	$\frac{1}{12}$
9	$\frac{1}{6}$

Tabelle 2: Inner products $\langle \phi(6), \phi(n) \rangle$ for selected integers

4.12.1 Example: Evaluating $\mu(\alpha)$ and $\mu(\neg\alpha)$

Let α correspond to $n = 8$. We compute:

$$t_\alpha = \langle \phi(6), \phi(8) \rangle = \frac{1}{12}$$

Since $t_\alpha > 0$, we have:

$$\mu(\alpha) = T$$

Similarly, for the negation:

$$\mu(\neg\alpha) = \mu(-t_\alpha) = \mu\left(-\frac{1}{12}\right) = F$$

4.12.2 Example: Logical Conjunction $\mu(\alpha \wedge \beta)$

Let β correspond to $n = 9$. We compute:

$$t_\beta = \langle \phi(6), \phi(9) \rangle = \frac{1}{6}$$

The conjunction is evaluated as:

$$t_{\alpha\wedge\beta} = \min\left(\frac{1}{12}, \frac{1}{6}\right) = \frac{1}{12}$$

$$\mu(\alpha \wedge \beta) = \mu\left(\frac{1}{12}\right) = T$$

4.12.3 Example: Logical Disjunction $\mu(\alpha \vee \beta)$

The disjunction is evaluated as:

$$t_{\alpha\vee\beta} = \max\left(\frac{1}{12}, \frac{1}{6}\right) = \frac{1}{6}$$

$$\mu(\alpha \vee \beta) = \mu\left(\frac{1}{6}\right) = T$$

4.12.4 Example: Logical Implication $\mu(\alpha \to \beta)$

The implication is evaluated as:

$$t_{\alpha\to\beta} = \max\left(-\frac{1}{12}, \frac{1}{6}\right) = \frac{1}{6}$$

$$\mu(\alpha \to \beta) = \mu\left(\frac{1}{6}\right) = T$$

4.13 Quantifiers over a Subset

We define the subset $M \subset X$ as the set of even integers in X:

$$M = \{n \in X \mid n \text{ is even}\} = \{-2, 2, 6, 8\}$$

4.13.1 Existential Quantifier $\exists n \in M,\ P(n)$

We compute the inner products for $n \in M$:

n	$t_n = \langle \phi(6), \phi(n) \rangle$
-2	$-\frac{1}{3}$
2	$\frac{1}{3}$
6	1
8	$\frac{1}{12}$

The maximum value is:

$$t_\exists = \max_{n \in M} t_n = 1$$

Since $t_\exists > 0$:

$$\mu\left(\exists n \in M,\ P(n)\right) = T$$

4.13.2 Universal Quantifier $\forall n \in M,\ P(n)$

The minimum value is:

$$t_\forall = \min_{n \in M} t_n = -\frac{1}{3}$$

Since $t_\forall < 0$:

$$\mu\left(\forall n \in M,\ P(n)\right) = F$$

4.13.3 Verification of De Morgan's Law for Quantifiers

We verify the equivalence:

$$\mu\left(\neg\left(\forall n \in M,\ P(n)\right)\right) = \mu\left(\exists n \in M,\ \neg P(n)\right)$$

Compute:

$$t_{\neg\forall} = -t_\forall = \frac{1}{3}$$

Since $t_{\neg\forall} > 0$:

$$\mu\left(\neg\left(\forall n \in M,\, P(n)\right)\right) = T$$

Compute the negation of $P(n)$ for each $n \in M$:

n	$-t_n$
-2	$\frac{1}{3}$
2	$-\frac{1}{3}$
6	-1
8	$-\frac{1}{12}$

The maximum of the negated values is:

$$t_{\exists\neg P} = \max_{n \in M}(-t_n) = \frac{1}{3}$$

Since $t_{\exists\neg P} > 0$:

$$\mu\left(\exists n \in M,\, \neg P(n)\right) = T$$

Thus, we have:

$$\mu\left(\neg\left(\forall n \in M,\, P(n)\right)\right) = \mu\left(\exists n \in M,\, \neg P(n)\right) = T$$

4.14 Interpretation of Results

The example computations demonstrate that:

- $\mu(\alpha) = T$ and $\mu(\neg\alpha) = F$ for α corresponding to $n = 8$.
- The logical conjunction $\mu(\alpha \wedge \beta) = T$ and disjunction $\mu(\alpha \vee \beta) = T$.
- The logical implication $\mu(\alpha \to \beta) = T$ holds.
- For the subset M, the existential quantifier evaluates to T and the universal quantifier evaluates to F.
- De Morgan's Law for quantifiers is verified, as both sides of the equivalence evaluate to T.

4.15 Application of the Semantic Space Framework

These examples demonstrate how the semantic space framework can be applied to integer embeddings. By representing integers as vectors in a high-dimensional space or directly using the kernel for the computations (Kernel-Trick) and defining logical operations in terms of inner products / kernel, we can explore logical relationships and properties in a geometric context.

4.16 Conclusion of Examples

We observe that classical logical properties, such as conjunction, disjunction, implication, and quantifiers, can be represented and evaluated within this framework. This approach offers a novel perspective on logical reasoning, bridging the gap between algebraic number theory and logical semantics.

5 Formalization of the Semantic Space of Logic

Definition 5.1 (Semantic Space)**.** A **semantic space** $S = (X, k)$ consists of:

- A set X.
- A positive semidefinite kernel function $k : X \times X \to \mathbb{R}$ satisfying:
  - $-1 \leq k(x, y) \leq 1$ for all $x, y \in X$.
  - $k(x, y) = 1$ if and only if $x = y$.

Definition 5.2 (Semantic Space of Logic)**.** A **semantic space of logic** $L = (S, \wedge, \vee, \neg, \to, \leftrightarrow, \forall, \exists)$ is defined over a semantic space $S = (X, k)$ with the following components:

- $\wedge$ (Conjunction): A binary operation on S that is commutative, associative, and distributive over $\vee$.
- $\vee$ (Disjunction): A binary operation on S that is commutative, associative, and distributes over $\wedge$.
- $\neg$ (Negation): A unary operation on S that is an involution, i.e., $\neg(\neg\alpha) = \alpha$ for all $\alpha \in S$.
- $\to$ (Implication): A binary operation on S satisfying contraposition, modus ponens, and other implication-related properties.
- $\leftrightarrow$ (Biconditional): A binary operation on S defined in terms of $\to$ and $\wedge$, satisfying properties such as symmetry and reflexivity.
- $\forall$ (Universal Quantifier): A quantifier that interacts with logical connectives, preserving logical equivalences and De Morgan's laws.
- $\exists$ (Existential Quantifier): A quantifier that interacts with logical connectives, preserving logical equivalences and De Morgan's laws.

Remark 5.3. The operations $\wedge$, $\vee$, $\neg$, $\to$, $\leftrightarrow$, $\forall$, and $\exists$ are defined abstractly without reference to specific mathematical constructs such as min, max, or scalar negation. Instead, they are characterized by the logical properties they satisfy, ensuring that all classical logical laws hold within the semantic space of logic L.

5.0.1 Properties of Logical Operations

The logical operations in the semantic space of logic L are designed to satisfy the classical logical properties as follows:

Commutativity

- $\alpha \wedge \beta = \beta \wedge \alpha$ for all $\alpha, \beta \in S$.
- $\alpha \vee \beta = \beta \vee \alpha$ for all $\alpha, \beta \in S$.

Associativity

- $(\alpha \wedge \beta) \wedge \gamma = \alpha \wedge (\beta \wedge \gamma)$ for all $\alpha, \beta, \gamma \in S$.
- $(\alpha \vee \beta) \vee \gamma = \alpha \vee (\beta \vee \gamma)$ for all $\alpha, \beta, \gamma \in S$.

Distributivity

- $\alpha \wedge (\beta \vee \gamma) = (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ for all $\alpha, \beta, \gamma \in S$.
- $\alpha \vee (\beta \wedge \gamma) = (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ for all $\alpha, \beta, \gamma \in S$.

De Morgan's Laws

- $\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$ for all $\alpha, \beta \in S$.
- $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$ for all $\alpha, \beta \in S$.

Double Negation

- $\neg(\neg\alpha) = \alpha$ for all $\alpha \in S$.

Modus Ponens

- If $\mu(\alpha) = T$ and $\mu(\alpha \to \beta) = T$, then $\mu(\beta) = T$.

Modus Tollens

- If $\mu(\alpha \to \beta) = T$ and $\mu(\beta) = F$, then $\mu(\alpha) = F$.

Contraposition

- $\alpha \to \beta = \neg\beta \to \neg\alpha$ for all $\alpha, \beta \in S$.

Identity of Implication

- $\mu(\alpha \to \alpha) \neq F$ for all $\alpha \in S$.

Disjunctive Syllogism

- If $\mu(\alpha \vee \beta) = T$ and $\mu(\neg\alpha) = T$, then $\mu(\beta) = T$.

Law of Excluded Middle

- $\mu(\alpha \vee \neg\alpha) \neq F$ for all $\alpha \in S$.

Law of Non-Contradiction

- $\mu(\alpha \wedge \neg\alpha) \neq T$ for all $\alpha \in S$.

5.0.2 Properties of Quantifiers

The quantifiers $\forall$ and $\exists$ interact with the logical connectives as follows:

Negation of Quantifiers

- $\neg(\forall a \in X,\, P(a)) = \exists a \in X,\, \neg P(a)$
- $\neg(\exists a \in X,\, P(a)) = \forall a \in X,\, \neg P(a)$

Distributivity over Logical Connectives

- $\forall a \in X,\, (P(a) \wedge Q(a)) = (\forall a \in X,\, P(a)) \wedge (\forall a \in X,\, Q(a))$
- $\exists a \in X,\, (P(a) \vee Q(a)) = (\exists a \in X,\, P(a)) \vee (\exists a \in X,\, Q(a))$

Interaction with Implication

- $(\forall a \in X,\, P(a)) \to Q = \exists a \in X,\, (P(a) \to Q)$
- $P \to (\forall a \in X,\, Q(a)) = \forall a \in X,\, (P \to Q(a))$

5.0.3 Evaluation Function

An evaluation function $\mu : S \to \{T, I, F\}$ assigns a truth value to each element of the semantic space based on its projection onto a fixed perspective vector w. Formally:

Definition 5.4 (Evaluation Function)**.** Let $w \in X$ be a fixed perspective vector. The evaluation function $\mu : S \to \{T, I, F\}$ is defined by:

$$\mu(\alpha) = \begin{cases} T & \text{if } \langle \phi(w), \alpha \rangle > 0, \\ I & \text{if } \langle \phi(w), \alpha \rangle = 0, \\ F & \text{if } \langle \phi(w), \alpha \rangle < 0. \end{cases}$$

Remark 5.5. The evaluation function abstracts the determination of truth values based on the geometric relationship between elements in the semantic space and the perspective vector w. It ensures that logical operations conform to the classical properties by leveraging the underlying geometric structure.

5.0.4 Ensuring Logical Properties

To guarantee that all logical properties hold within the semantic space of logic L, the operations $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$, $\forall$, and $\exists$ are defined to satisfy the following axioms:

- **Commutativity**: Operations $\wedge$ and $\vee$ are commutative.
- **Associativity**: Operations $\wedge$ and $\vee$ are associative.
- **Distributivity**: Operations $\wedge$ distributes over $\vee$ and vice versa.
- **De Morgan's Laws**: Negation distributes over $\wedge$ and $\vee$.
- **Double Negation**: Negation is an involution.
- **Implication Properties**: Implication satisfies contraposition, modus ponens, and modus tollens.
- **Biconditional Properties**: Biconditional is symmetric and reflexive.
- **Quantifier Properties**: Quantifiers interact with logical connectives according to classical logic, including distributivity and negation.

These axioms abstractly define the behavior of logical operations without relying on specific mathematical constructs like min, max, or scalar negation. Instead, they ensure that the operations preserve the classical logical structure within the geometric framework of the semantic space.

Theorem 5.6 (One example semantic space of logic)**.** *Let L be as defined in the beginning of this document ($\min, \max, \neg a = -a$). Then L is a semantic space of logic.*

Beweis. The proof of this statement has been given by providing proofs in this document of the individual properties. □

Remark 5.7. This formalization abstracts the logical operations to their essential properties, allowing the semantic space framework to model logical reasoning geometrically. By adhering to the classical logical axioms, the framework ensures compatibility with traditional logical systems while leveraging the advantages of geometric representations.

6 The Lukasiewicz semantic space of logic

To prove that the Lukasiewicz semantic space of logic $L := ((X,k), \wedge, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists, \mu)$ satisfies all properties that make it a semantic space of logic, we need to verify the following logical properties for the operations defined in this space:

6.1 Lukasiewicz Operations in the Semantic Space

Given:

- $S = (X, k)$, where $0 \leq k(x, y) \leq 1$ and $k(x, y) = 1$ iff $x = y \ \forall x, y \in X$ and $k : X \times X \to \mathbb{R}$ is positive semidefinite.
- Definition of logical operations:
  - Conjunction: $a \wedge b := \min(a, b)$.
  - Disjunction: $a \vee b := \max(a, b)$.
  - Negation: $\neg a := 1 - a$.
  - Implication: $a \to b := \min(1, 1 - a + b)$.
  - Biconditional: $a \leftrightarrow b := 1 - |a - b|$.
  - Universal quantifier: $\forall x \in M, P(x) := \min_{x \in M} \langle \phi(w), P(x) \rangle$.
  - Existential quantifier: $\exists x \in M, P(x) := \max_{x \in M} \langle \phi(w), P(x) \rangle$.
- Evaluation function μ:
  - $\mu(a) = T$, if $a > 0.5$.
  - $\mu(a) = I$, if $a = 0.5$.
  - $\mu(a) = F$, if $a < 0.5$.

6.2 Step-by-Step Proof of Properties

6.2.1 1. Commutativity

- **Conjunction**: $\min(a, b) = \min(b, a)$.
- **Disjunction**: $\max(a, b) = \max(b, a)$.

Both operations are commutative by the definition of min and max functions.

6.2.2 2. Associativity

- **Conjunction**: $(a \wedge b) \wedge c = \min(\min(a, b), c) = \min(a, b, c) = \min(a, \min(b, c)) = a \wedge (b \wedge c)$.
- **Disjunction**: $(a \vee b) \vee c = \max(\max(a, b), c) = \max(a, b, c) = \max(a, \max(b, c)) = a \vee (b \vee c)$.

Both operations are associative by the properties of min and max.

6.2.3 3. Distributivity

Conjunction over Disjunction:

$$a \wedge (b \vee c) = \min(a, \max(b, c)), \quad (\min(a, b) \vee \min(a, c)) = \max(\min(a, b), \min(a, c)).$$

- If $a \leq \max(b, c)$, then $a = \min(a, \max(b, c)) = \max(\min(a, b), \min(a, c))$.
- If $a \geq \max(b, c)$, then $\min(a, \max(b, c)) = \max(b, c)$ and $\max(\min(a, b), \min(a, c)) = \max(b, c)$.

Thus, distributivity holds.

6.2.4 4. Negation

- **Double Negation**: $\neg(\neg a) = 1 - (1 - a) = a$.

6.2.5 5. Implication Properties

- **Contraposition**: $a \to b = \min(1, 1 - a + b) = \min(1, 1 - b + a) = \neg b \to \neg a$.
- **Modus Ponens**:
  - Given $\mu(a) = T$ (i.e., $a > 0.5$) and $\mu(a \to b) = T$ (i.e., $\min(1, 1-a+b) > 0.5$).
  - Since $a > 0.5$, $1 - a + b > 0.5$, implying $b > 0.5$, so $\mu(b) = T$.
- **Modus Tollens**:
  - Given $\mu(a \to b) = T$ and $\mu(b) = F$.
  - Since $b < 0.5$, $1 - a + b > 0.5$ must hold, implying $1 - a > 0.5$, or $a < 0.5$, so $\mu(a) = F$.

6.2.6 6. Biconditional Properties

- **Symmetry**: $a \leftrightarrow b = 1 - |a - b| = 1 - |b - a| = b \leftrightarrow a$.
- **Reflexivity**: $a \leftrightarrow a = 1 - |a - a| = 1$.

6.2.7 7. Quantifiers

- **Negation of Universal Quantification**:

$$\neg (\forall x \in M, P(x)) = 1 - \min_{x \in M} \langle \phi(w), P(x) \rangle = \max_{x \in M} (1 - \langle \phi(w), P(x) \rangle) = \exists x \in M, \neg P(x).$$

- **Negation of Existential Quantification**:

$$\neg (\exists x \in M, P(x)) = 1 - \max_{x \in M} \langle \phi(w), P(x) \rangle = \min_{x \in M} (1 - \langle \phi(w), P(x) \rangle) = \forall x \in M, \neg P(x).$$

6.2.8 8. Law of Excluded Middle

$$a \vee \neg a = \max(a, 1-a) = 1, \quad \mu(a \vee \neg a) = T.$$

The Law of Excluded Middle holds since the maximum of a and $1-a$ is always 1.

6.2.9 9. Law of Non-Contradiction

$$a \wedge \neg a = \min(a, 1-a) \leq 0.5, \quad \mu(a \wedge \neg a) = F.$$

The Law of Non-Contradiction holds because the minimum of a and $1-a$ is always less than or equal to 0.5.

6.3 Conclusion

All the classical logical properties hold within the Lukasiewicz space L. Therefore, $L := ((X,k), \wedge, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists, \mu)$ is a valid **Lukasiewicz semantic space of logic**.

7 Example: Evaluating Hypotheses in the Lukasiewicz Semantic Space

7.1 Toy dataset description

Consider the toy dataset of **smartphones** with properties such as 'Battery Life', 'Screen Size', 'Weight', 'Performance', and 'Price'. The kernel for these properties is given by the following Gram matrix G:

$$G = \begin{array}{c|ccccc} & \text{Battery Life} & \text{Screen Size} & \text{Weight} & \text{Performance} & \text{Price} \\ \hline \text{Battery Life} & 1.00 & \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\ \text{Screen Size} & \frac{1}{2} & 1.00 & \frac{1}{12} & \frac{1}{6} & \frac{1}{3} \\ \text{Weight} & \frac{1}{6} & \frac{1}{12} & 1.00 & \frac{1}{2} & \frac{1}{4} \\ \text{Performance} & \frac{1}{3} & \frac{1}{6} & \frac{1}{2} & 1.00 & \frac{1}{2} \\ \text{Price} & \frac{1}{6} & \frac{1}{3} & \frac{1}{4} & \frac{1}{2} & 1.00 \end{array}$$

7.2 Feature Map and Perspective Vector

Perform a 'Cholesky Decomposition' on the Gram matrix G to find the embedding $\phi(x)$ such that:

$$G_{x,y} = k(x,y) = \langle \phi(x), \phi(y) \rangle$$

Select 'Performance' as the perspective vector w. Compute the projections of each property onto w:

$$\mathrm{proj}(w, x) = \langle \phi(w), \phi(x) \rangle = k(w, x)$$

7.3 Computed Projections

- **Battery Life**: $\mathrm{proj}(w, \text{Battery Life}) = \frac{1}{3} \approx 0.333 \to F$
- **Screen Size**: $\mathrm{proj}(w, \text{Screen Size}) = \frac{1}{6} \approx 0.167 \to F$
- **Weight**: $\mathrm{proj}(w, \text{Weight}) = \frac{1}{2} = 0.5 \to I$
- **Performance**: $\mathrm{proj}(w, \text{Performance}) = 1.00 \to T$
- **Price**: $\mathrm{proj}(w, \text{Price}) = \frac{1}{2} = 0.5 \to I$

These projections represent the degrees of association of each property with 'Performance' within the semantic space.

7.4 Formulating and Evaluating Hypotheses

Based on the dataset and the computed projections, we formulate several hypotheses in both propositional and first-order logic. Each hypothesis is then evaluated within the Lukasiewicz semantic space to determine its truth value (T, I, or F).

7.4.1 Propositional Logic Hypotheses

Hypothesis A :

$$\text{Performance} \to \text{Battery Life}$$

- **Meaning**: If a smartphone has high performance, then it has long battery life.
- **Evaluation**:

$$\text{Performance} \to \text{Battery Life} = \min\left(1, 1 - 1.00 + \frac{1}{3}\right) = \min\left(1, \frac{1}{3}\right) = \frac{1}{3} \approx 0.333$$

$$\mu\left(\text{Performance} \to \text{Battery Life}\right) = F \quad \text{since } 0.333 < \frac{1}{2}$$

- **Interpretation**: The hypothesis is **False**, indicating that higher performance does not necessarily imply longer battery life.

Hypothesis B :

$$\text{Price} \to (\text{Performance} \wedge \text{Screen Size})$$

- **Meaning**: If a smartphone is expensive, then it has high performance and a large screen size.
- **Evaluation**:

$$\text{Performance} \wedge \text{Screen Size} = \min\left(1.00, \frac{1}{6}\right) = \frac{1}{6} \approx 0.167$$

$$\text{Price} \to (\text{Performance}\wedge\text{Screen Size}) = \min\left(1, 1 - \frac{1}{2} + \frac{1}{6}\right) = \min\left(1, \frac{2}{3}\right) = \frac{2}{3} \approx 0.667$$

$$\mu\,(\text{Price} \to (\text{Performance} \wedge \text{Screen Size})) = T \quad \text{since } 0.667 > \frac{1}{2}$$

- **Interpretation**: The hypothesis is **True**, suggesting that more expensive smartphones tend to have both high performance and larger screen sizes.

Hypothesis C :

$$\text{Weight} \vee \text{Price}$$

- **Meaning**: The smartphone is either heavy or expensive.
- **Evaluation**:

$$\text{Weight} \vee \text{Price} = \max\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2} = 0.5$$

$$\mu\,(\text{Weight} \vee \text{Price}) = I \quad \text{since } 0.5 = \frac{1}{2}$$

- **Interpretation**: The hypothesis is **Indeterminate**, indicating that the disjunction is exactly at the threshold and cannot be conclusively determined as true or false.

7.4.2 First-Order Logic Hypotheses

Hypothesis D :

$$\forall a \in X, \, (\text{Performance} \to a)$$

- **Meaning**: For every property a, if a smartphone has high performance, then it possesses property a.
- **Evaluation**:

$$\forall a \in X, \, \text{Performance} \to a = \min\,(\text{Performance} \to \text{Battery Life}, \, \text{Performance} \to \text{Screen Size}, \, \text{Per}$$

Calculating each implication:

$$\text{Performance} \to \text{Battery Life} = \frac{1}{3} \approx 0.333$$

$$\text{Performance} \to \text{Screen Size} = \min\left(1, 1 - 1.00 + \frac{1}{6}\right) = \min\left(1, \frac{1}{6}\right) = \frac{1}{6} \approx 0.167$$

$$\text{Performance} \to \text{Weight} = \min\left(1, 1 - 1.00 + \frac{1}{2}\right) = \min\left(1, \frac{1}{2}\right) = \frac{1}{2} = 0.5$$

$$\text{Performance} \to \text{Performance} = \min(1, 1 - 1.00 + 1.00) = \min(1, 1) = 1.00$$

$$\text{Performance} \to \text{Price} = \min\left(1, 1 - 1.00 + \frac{1}{2}\right) = \min\left(1, \frac{1}{2}\right) = \frac{1}{2} = 0.5$$

Therefore:

$$\forall a \in X,\ \text{Performance} \to a = \min(0.333, 0.167, 0.5, 1.00, 0.5) = 0.167$$

$$\mu(\forall a \in X,\ \text{Performance} \to a) = F \quad \text{since } 0.167 < \frac{1}{2}$$

- **Interpretation**: The universal statement is **False** because there exists at least one property ('Screen Size') for which the implication Performance $\to a$ is false.

Hypothesis E :

$$\exists a \in X,\ (\text{Price} \to a)$$

- **Meaning**: There exists at least one property a such that if a smartphone is expensive, then it possesses property a.
- **Evaluation**:

$$\exists a \in X,\ \text{Price} \to a = \max(\text{P.} \to \text{Bt.L.}, \text{P.} \to \text{Scr.Si.}, \text{P.} \to \text{W.}, \text{P.} \to \text{Perf.}, \text{P.} \to \text{P.})$$

Calculating each implication:

$$\text{Price} \to \text{Battery Life} = \min\left(1, 1 - \frac{1}{2} + \frac{1}{3}\right) = \min\left(1, \frac{5}{6}\right) = \frac{5}{6} \approx 0.833$$

$$\text{Price} \to \text{Screen Size} = \min\left(1, 1 - \frac{1}{2} + \frac{1}{6}\right) = \min\left(1, \frac{2}{3}\right) = \frac{2}{3} \approx 0.667$$

$$\text{Price} \to \text{Weight} = \min\left(1, 1 - \frac{1}{2} + \frac{1}{4}\right) = \min\left(1, \frac{3}{4}\right) = \frac{3}{4} = 0.75$$

$$\text{Price} \to \text{Performance} = \min\left(1, 1 - \frac{1}{2} + 1.00\right) = \min\left(1, \frac{3}{2}\right) = 1.00$$

$$\text{Price} \to \text{Price} = \min\left(1, 1 - \frac{1}{2} + \frac{1}{2}\right) = \min(1, 1) = 1.00$$

Therefore:

$$\exists a \in X, \text{Price} \to a = \max(0.833, 0.667, 0.75, 1.00, 1.00) = 1.00$$

$$\mu(\exists a \in X, \text{Price} \to a) = T \quad \text{since } 1.00 > \frac{1}{2}$$

- **Interpretation**: The existential statement is **True** because there exists at least one property ('Performance' and 'Price' themselves) for which the implication Price $\to a$ holds true.

Hypothesis F :

$$\neg(\forall a \in X, \text{Performance} \to a)$$

- **Meaning**: It is not the case that for every property a, if a smartphone has high performance, then it possesses property a.

- **Evaluation**:

$$\neg(\forall a \in X, \text{Performance} \to a) = \exists a \in X, \neg(\text{Performance} \to a)$$

Calculate $\neg(\text{Performance} \to a)$ for each a:

$$\neg(\text{Performance} \to a) = 1 - (\text{Performance} \to a) = 1 - \min(1, 1 - 1.00 + a)$$

Since Performance $= 1.00$, this simplifies to:

$$\neg(\text{Performance} \to a) = 1 - \min(1, a) = \max(0, 1 - a)$$

Evaluating for each property:

$$\begin{aligned}
\text{Battery Life}: &\quad \neg(\text{Performance} \to \text{Battery Life}) = 1 - 0.333 = 0.667 \\
\text{Screen Size}: &\quad \neg(\text{Performance} \to \text{Screen Size}) = 1 - 0.167 = 0.833 \\
\text{Weight}: &\quad \neg(\text{Performance} \to \text{Weight}) = 1 - 0.5 = 0.5 \\
\text{Performance}: &\quad \neg(\text{Performance} \to \text{Performance}) = 1 - 1.00 = 0.0 \\
\text{Price}: &\quad \neg(\text{Performance} \to \text{Price}) = 1 - 0.5 = 0.5
\end{aligned}$$

Therefore:

$$\exists a \in X, \neg(\text{Performance} \to a) = \max(0.667, 0.833, 0.5, 0.0, 0.5) = 0.833$$

$$\mu(\neg(\forall a \in X, \text{Performance} \to a)) = T \quad \text{since } 0.833 > \frac{1}{2}$$

- **Interpretation**: The negation of the universal statement is **True**, reinforcing that not all implications from performance to other properties are valid.

7.4.3 Summary of Hypotheses Evaluation

Hypothesis	Logical Form	Evaluation	Truth Value
A	Performance $\rightarrow$ Battery Life	$0.333 < 0.5$	F
B	Price $\rightarrow$ (Performance $\wedge$ Screen Size)	$0.667 > 0.5$	T
C	Weight $\vee$ Price	$0.5 = 0.5$	I
D	$\forall a \in X$, Performance $\rightarrow a$	$0.167 < 0.5$	F
E	$\exists a \in X$, Price $\rightarrow a$	$1.00 > 0.5$	T
F	$\neg(\forall a \in X$, Performance $\rightarrow a)$	$0.833 > 0.5$	T

Tabelle 3: Summary of Hypotheses Evaluation in the Lukasiewicz Semantic Space

Overall Interpretation:

- **Hypothesis A** is **False**, indicating that higher performance does not necessarily imply longer battery life. This aligns with real-world observations where high-performance devices often consume more power, reducing battery life.
- **Hypothesis B** is **True**, suggesting that more expensive smartphones tend to have both high performance and larger screen sizes. This matches consumer expectations that premium devices offer superior features.
- **Hypothesis C** is **Indeterminate**, indicating that the disjunction is exactly at the threshold. This reflects a balanced relationship where being heavy or expensive is neither strongly true nor false.
- **Hypothesis D** is **False**, highlighting that not all properties are implied by performance. Specifically, 'Screen Size' does not follow from high performance, which is realistic as performance improvements do not necessarily require larger screens.
- **Hypothesis E** is **True**, confirming that there exists at least one property ('Performance' itself) for which the implication holds true. This is intuitive, as a device's performance inherently relates to itself.
- **Hypothesis F** is **True**, reinforcing that it is not the case that all properties are implied by performance. This underscores the selective influence of performance on other device attributes.

These evaluations demonstrate the effectiveness of the Lukasiewicz semantic space in validating logical hypotheses against a realistic dataset. By leveraging the defined logical operations and the evaluation function, we can rigorously test and interpret the relationships between different properties within the semantic space framework.

7.5 Conclusion of the Example

This example illustrates how the Lukasiewicz semantic space of logic can be applied to a practical dataset of smartphones. By defining logical operations within the semantic space and utilizing the evaluation function, we successfully formulated and evaluated multiple hypotheses, uncovering meaningful relationships between smartphone properties. This approach showcases the potential of semantic spaces in modeling and reasoning about complex, real-world data, providing a robust framework for logical analysis and knowledge representation.

8 Toy Example: Evaluating Hypotheses About Animals (Cats) Using the Lukasiewicz Semantic Space Framework

8.1 Assigned Animals and Similarity Matrix

8.1.1 Animals

1. **Domestic Cat**
2. **Tiger**
3. **Lion**
4. **Cheetah**
5. **Leopard**

8.1.2 Similarity Matrix

	Domestic Cat	**Tiger**	**Lion**	**Cheetah**	**Leopard**
Domestic Cat	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
Tiger	$\frac{1}{2}$	1	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{1}{3}$
Lion	$\frac{2}{3}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{4}$
Cheetah	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{2}$	1	$\frac{1}{2}$
Leopard	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{2}$	1

Tabelle 4: Similarity Matrix Among Selected Felines

8.1.3 Interpretation

- **Similarity Scores** range from 1 (identical) to lower fractions indicating lesser similarity.
- The diagonal elements are all 1, representing perfect similarity of an animal with itself.

Abbildung 2: Cats

8.2 Lukasiewicz Semantic Space Framework

8.2.1 Definitions Recap

- **Conjunction** ($\wedge$): $\alpha \wedge \beta = \min(\alpha, \beta)$
- **Disjunction** ($\vee$): $\alpha \vee \beta = \max(\alpha, \beta)$
- **Negation** ($\neg$): $\neg\alpha = 1 - \alpha$
- **Implication** ($\rightarrow$): $\alpha \rightarrow \beta = \min(1, 1 - \alpha + \beta)$
- **Biconditional** ($\leftrightarrow$): $\alpha \leftrightarrow \beta = 1 - |\alpha - \beta|$
- **Evaluation Function** (μ):

$$\mu(\alpha) = \begin{cases} T & \text{if } \alpha > 0.5, \\ I & \text{if } \alpha = 0.5, \\ F & \text{if } \alpha < 0.5. \end{cases}$$

8.2.2 Perspective Vector

For consistency, we select **Domestic Cat** as our perspective vector w. Thus, the projection of each animal onto w is given by their similarity score.

8.3 Formulating Hypotheses

We will formulate the following hypotheses about the felines and evaluate them within the Lukasiewicz semantic space:

1. **Hypothesis A**: *If a Domestic Cat is similar to a Tiger, then it is similar to a Lion.*
2. **Hypothesis B**: *A Domestic Cat is either similar to a Cheetah or not similar to a Leopard.*
3. **Hypothesis C**: *A Domestic Cat is similar to a Tiger and a Lion.*
4. **Hypothesis D**: *For every pair of animals, if one is similar to a Cheetah, then the other is not similar to a Leopard.*
5. **Hypothesis E**: *There exists at least one animal that is neither similar to a Domestic Cat nor similar to a Tiger.*

8.4 Detailed Evaluation of Hypotheses

8.4.1 Hypothesis A: If a Domestic Cat is similar to a Tiger, then it is similar to a Lion.

Logical Formulation

$$\text{Tiger} \to \text{Lion}$$

Evaluation

$$\text{Tiger} = \frac{1}{2}, \quad \text{Lion} = \frac{2}{3}$$

$$\text{Tiger} \to \text{Lion} = \min\left(1, 1 - \frac{1}{2} + \frac{2}{3}\right) = \min\left(1, \frac{7}{6}\right) = 1$$

$$\mu\left(\text{Tiger} \to \text{Lion}\right) = T$$

Interpretation The implication evaluates to **True**. This suggests that within the semantic space, the similarity of Domestic Cat to Tiger positively influences its similarity to Lion.

8.4.2 Hypothesis B: A Domestic Cat is either similar to a Cheetah or not similar to a Leopard.

Logical Formulation

$$\text{Cheetah} \vee \neg\text{Leopard}$$

Evaluation

$$\text{Cheetah} = \frac{1}{3}, \quad \neg\text{Leopard} = 1 - \frac{1}{6} = \frac{5}{6}$$
$$\text{Cheetah} \vee \neg\text{Leopard} = \max\left(\frac{1}{3}, \frac{5}{6}\right) = \frac{5}{6}$$
$$\mu\left(\text{Cheetah} \vee \neg\text{Leopard}\right) = T$$

Interpretation The disjunction evaluates to **True**, indicating that a Domestic Cat is either somewhat similar to a Cheetah or significantly dissimilar to a Leopard.

8.4.3 Hypothesis C: A Domestic Cat is similar to a Tiger and a Lion.

Logical Formulation

$$\text{Tiger} \wedge \text{Lion}$$

Evaluation

$$\text{Tiger} = \frac{1}{2}, \quad \text{Lion} = \frac{2}{3}$$
$$\text{Tiger} \wedge \text{Lion} = \min\left(\frac{1}{2}, \frac{2}{3}\right) = \frac{1}{2}$$
$$\mu\left(\text{Tiger} \wedge \text{Lion}\right) = I$$

Interpretation The conjunction evaluates to **Indeterminate**. This reflects a balanced similarity where Domestic Cat shares moderate similarity with both Tiger and Lion, but not strongly enough to be definitively true.

8.4.4 Hypothesis D: For every pair of animals, if one is similar to a Cheetah, then the other is not similar to a Leopard.

Logical Formulation

$$\forall x \in \{\text{Domestic Cat}, \text{Tiger}, \text{Lion}, \text{Cheetah}, \text{Leopard}\},\ (x \to \neg\text{Leopard})$$

Evaluation For each animal x, compute $x \to \neg\text{Leopard}$:

1. **Domestic Cat**:
$$x = 1, \quad \neg\text{Leopard} = 1 - \frac{1}{6} = \frac{5}{6}$$
$$x \to \neg\text{Leopard} = \min\left(1, 1 - 1 + \frac{5}{6}\right) = \min\left(1, \frac{5}{6}\right) = \frac{5}{6}$$

2. **Tiger**:
$$x = \frac{1}{2}, \quad \neg\text{Leopard} = \frac{5}{6}$$
$$x \to \neg\text{Leopard} = \min\left(1, 1 - \frac{1}{2} + \frac{5}{6}\right) = \min\left(1, \frac{7}{6}\right) = 1$$

3. **Lion**:
$$x = \frac{2}{3}, \quad \neg\text{Leopard} = \frac{5}{6}$$
$$x \to \neg\text{Leopard} = \min\left(1, 1 - \frac{2}{3} + \frac{5}{6}\right) = \min\left(1, \frac{7}{6}\right) = 1$$

4. **Cheetah**:
$$x = \frac{1}{3}, \quad \neg\text{Leopard} = \frac{5}{6}$$
$$x \to \neg\text{Leopard} = \min\left(1, 1 - \frac{1}{3} + \frac{5}{6}\right) = \min\left(1, \frac{7}{6}\right) = 1$$

5. **Leopard**:
$$x = \frac{1}{6}, \quad \neg\text{Leopard} = \frac{5}{6}$$
$$x \to \neg\text{Leopard} = \min\left(1, 1 - \frac{1}{6} + \frac{5}{6}\right) = \min(1, 1) = 1$$

Therefore:
$$\forall x,\ x \to \neg\text{Leopard} = \min\left(\frac{5}{6}, 1, 1, 1, 1\right) = \frac{5}{6}$$
$$\mu(\forall x,\ x \to \neg\text{Leopard}) = T$$

Interpretation The universal statement evaluates to **True**, indicating that for every animal, if it is similar to a Cheetah, then it is not significantly similar to a Leopard. This aligns with biological distinctions where Cheetahs and Leopards occupy different ecological niches.

8.4.5 Hypothesis E: There exists at least one animal that is neither similar to a Domestic Cat nor similar to a Tiger.

Logical Formulation

$$\exists x \in \{\text{Lion}, \text{Cheetah}, \text{Leopard}\},\ (\neg\text{Domestic Cat} \land \neg\text{Tiger})$$

Evaluation First, identify animals that are neither similar to Domestic Cat nor to Tiger:

1. **Lion**:

$$\text{Similarity with Domestic Cat} = \frac{2}{3}, \quad \text{Similarity with Tiger} = \frac{3}{4}$$

$$\neg\text{Domestic Cat} = 1 - \frac{2}{3} = \frac{1}{3}, \quad \neg\text{Tiger} = 1 - \frac{3}{4} = \frac{1}{4}$$

$$\neg\text{Domestic Cat} \land \neg\text{Tiger} = \min\left(\frac{1}{3}, \frac{1}{4}\right) = \frac{1}{4}$$

2. **Cheetah**:

$$\text{Similarity with Domestic Cat} = \frac{1}{3}, \quad \text{Similarity with Tiger} = \frac{2}{3}$$

$$\neg\text{Domestic Cat} = 1 - \frac{1}{3} = \frac{2}{3}, \quad \neg\text{Tiger} = 1 - \frac{2}{3} = \frac{1}{3}$$

$$\neg\text{Domestic Cat} \land \neg\text{Tiger} = \min\left(\frac{2}{3}, \frac{1}{3}\right) = \frac{1}{3}$$

3. **Leopard**:

$$\text{Similarity with Domestic Cat} = \frac{1}{6}, \quad \text{Similarity with Tiger} = \frac{1}{3}$$

$$\neg\text{Domestic Cat} = 1 - \frac{1}{6} = \frac{5}{6}, \quad \neg\text{Tiger} = 1 - \frac{1}{3} = \frac{2}{3}$$

$$\neg\text{Domestic Cat} \land \neg\text{Tiger} = \min\left(\frac{5}{6}, \frac{2}{3}\right) = \frac{2}{3}$$

Therefore:

$$\exists x,\ (\neg\text{Domestic Cat} \land \neg\text{Tiger}) = \max\left(\frac{1}{4}, \frac{1}{3}, \frac{2}{3}\right) = \frac{2}{3}$$

$$\mu\left(\exists x,\ (\neg\text{Domestic Cat} \land \neg\text{Tiger})\right) = T$$

Interpretation The existential statement evaluates to **True**, indicating that there exists at least one animal (e.g., Leopard) that is neither significantly similar to a Domestic Cat nor to a Tiger.

8.5 Summary of Hypotheses Evaluation

Hypothesis	Logical Form	Evaluation	Truth Value
A	Tiger $\rightarrow$ Lion	1	**T**
B	Cheetah $\vee$ $\neg$Leopard	$\frac{5}{6}$	**T**
C	Tiger $\wedge$ Lion	$\frac{1}{2}$	**I**
D	$\forall x,\ x \rightarrow \neg$Leopard	$\frac{5}{6}$	**T**
E	$\exists x,\ (\neg$Domestic Cat $\wedge$ $\neg$Tiger$)$	$\frac{2}{3}$	**T**

Tabelle 5: Summary of Hypotheses Evaluation

8.6 Interpretation of Results

1. **Hypothesis A (True)**:
   - **Conclusion**: If a Domestic Cat is similar to a Tiger, it is indeed similar to a Lion.
   - **Insight**: This aligns with biological taxonomy, as Tigers and Lions are both big cats within the same genus.
2. **Hypothesis B (True)**:
   - **Conclusion**: A Domestic Cat is either somewhat similar to a Cheetah or significantly dissimilar to a Leopard.
   - **Insight**: Domestic Cats share moderate similarities with Cheetahs but are less similar to Leopards, reflecting differences in behavior and habitat.
3. **Hypothesis C (Indeterminate)**:
   - **Conclusion**: The Domestic Cat's similarity to both Tiger and Lion is balanced, neither definitively true nor false.
   - **Insight**: While Domestic Cats share traits with both Tigers and Lions, the moderate similarity suggests a nuanced relationship.
4. **Hypothesis D (True)**:
   - **Conclusion**: For every animal, if it is similar to a Cheetah, then it is not significantly similar to a Leopard.
   - **Insight**: Cheetahs and Leopards occupy different ecological niches, and their dissimilarities are reflected in the semantic space.
5. **Hypothesis E (True)**:

- **Conclusion**: There exists at least one animal (e.g., Leopard) that is neither significantly similar to a Domestic Cat nor to a Tiger.
- **Insight**: This highlights the diversity within the feline family, where certain species like Leopards exhibit distinct characteristics.

8.7 Visual Representation of Hypotheses

To further elucidate the relationships, consider the following simplified visualization:

- **Domestic Cat**:
  - Strong similarity with itself (1).
  - Moderate similarity with Tiger (0.5) and Lion (0.666).
  - Lower similarity with Cheetah (0.333) and Leopard (0.166).
- **Tiger**:
  - High similarity with Lion (0.75).
  - Moderate similarity with Cheetah (0.666) and Leopard (0.333).
- **Lion**:
  - Moderate similarity with Cheetah (0.5) and lower with Leopard (0.25).
- **Cheetah**:
  - Moderate similarity with Leopard (0.5).

This structure illustrates how the semantic space captures varying degrees of similarity among the felines, enabling the formulation and evaluation of logical hypotheses.

8.8 Conclusion

By leveraging the **Lukasiewicz semantic space** framework, we have successfully formulated and evaluated logical hypotheses about the similarities among various felines. This method provides a robust mechanism for reasoning about relationships within a set of entities based on their similarity metrics. The framework's ability to handle indeterminate truth values further enhances its applicability in scenarios where relationships are nuanced and not strictly binary.

This approach can be extended to more complex datasets and broader sets of entities, offering valuable insights in fields such as biology, taxonomy, and cognitive science. Additionally, integrating this framework with machine learning models can facilitate automated reasoning and knowledge discovery based on similarity measures.

9 Possible Applications

The semantic space framework described in this document has the potential to benefit society in multiple ways across various domains. By integrating logical reasoning with geometric representations, this system offers a novel approach to handling complex information, reasoning under uncertainty, and modeling human cognition.

9.1 Artificial Intelligence and Machine Learning

In the field of artificial intelligence (AI) and machine learning, the semantic space framework can enhance natural language processing (NLP) and understanding. By representing logical operations and quantifiers within a geometric space, AI systems can better interpret and generate human language that involves nuanced logical relationships.

- **Knowledge Representation**: The framework allows for the representation of knowledge in a continuous space, facilitating the handling of ambiguous or uncertain information.
- **Reasoning Under Uncertainty**: Incorporating the indeterminate truth value I enables AI systems to reason effectively even when information is incomplete or ambiguous.
- **Semantic Understanding**: Enhanced modeling of semantic relationships can improve machine translation, sentiment analysis, and information retrieval.

9.2 Cognitive Science and Psychology

The geometric approach to logic mirrors certain aspects of human cognition. By modeling how concepts relate in a semantic space, researchers can gain insights into how people process information and reason about the world.

- **Concept Formation**: Understanding how concepts are structured and related can inform theories of learning and memory.
- **Perspectivism**: The use of perspective vectors aligns with the idea that individuals have different viewpoints, which can be modeled and analyzed.

9.3 Decision-Making Systems

In complex decision-making scenarios, especially those involving multiple criteria and stakeholders, the framework can assist in evaluating options based on various perspectives.

- **Multi-Criteria Decision Analysis**: Quantifying and comparing different criteria within a unified space can aid in making balanced decisions.
- **Conflict Resolution**: Modeling different viewpoints can help identify common ground and facilitate negotiations.

9.4 Semantic Web and Ontologies

The semantic space framework can enhance the development of ontologies and the semantic web by providing a mathematical foundation for representing and reasoning about knowledge.

- **Ontology Alignment**: Geometric representations can assist in aligning different ontologies by measuring the similarity of concepts.
- **Knowledge Integration**: Combining information from various sources becomes more manageable within a unified semantic space.

9.5 Philosophical Logic and Formal Epistemology

The framework offers a new perspective on traditional logical systems, potentially contributing to discussions in philosophical logic.

- **Non-Classical Logics**: Exploring how classical logical properties are preserved or modified can lead to the development of new logical systems.
- **Epistemic Logic**: Modeling knowledge and belief within a geometric space can provide insights into epistemological questions.

9.6 Education and Pedagogy

The intuitive geometric interpretation of logical concepts can serve as an educational tool.

- **Teaching Logic**: Visual representations can make abstract logical concepts more accessible to students.
- **Critical Thinking Skills**: Encouraging students to think about information from different perspectives enhances critical thinking.

9.7 Enhancing Communication and Understanding

By modeling different perspectives and their logical interactions, the framework can aid in improving communication across diverse groups.

- **Cross-Cultural Understanding**: Representing concepts from various cultural perspectives can foster mutual understanding.
- **Collaborative Problem-Solving**: Identifying and reconciling different viewpoints can lead to more effective collaboration.

9.8 Data Analysis and Interpretation

In fields such as data science and analytics, the framework can be used to interpret complex datasets.

- **Pattern Recognition**: Geometric representations can help identify patterns and relationships within data.
- **Uncertainty Quantification**: Incorporating indeterminate values allows for more nuanced analysis of uncertain or incomplete data.

9.9 Conclusion of Applications

Overall, the semantic space framework provides a versatile tool for modeling and reasoning about complex information. Its ability to integrate logical operations within a geometric context opens up new possibilities for AI, cognitive science, decision-making, and beyond. By embracing this approach, society can develop more sophisticated systems that handle ambiguity and diversity of perspectives, ultimately leading to more effective solutions to complex problems.

10 Application to the House Price Prediction Dataset from Kaggle

The House Price Prediction Dataset on Kaggle is a popular dataset used for predictive modeling, especially in regression tasks where the goal is to estimate house prices based on a variety of features. This dataset contains information about houses that cover a broad spectrum of attributes such as location, size, number of rooms, condition, and year built. It is frequently used in competitions and academic work to benchmark models for real estate price prediction.

10.1 Interpretation of Sequences of Properties

Let $S = (X, k)$ be a 'the' Lukasiewicz semantic space of logic derived below with the Python code for the house price prediction dataset and $x_1, x_2, \cdots, x_r$ be a sequence of elements of X, where X are the house properties. We want to be able to express the following idea as a formula: Starting from x_1 we deduce in a sequence of r steps that x_r must be true. Therefore we interpret this sequence as the following formula, which basically says, I start with x_1 as being true, then I keep on adding that from $x_1 \to x_i$, until we arrive at x_r:

$$F = x_1 \wedge (x_1 \to x_2) \wedge (x_2 \to x_3) \wedge \ldots \wedge (x_{r-1} \to x_r) \qquad (1)$$

This approach makes sense becaus Modus Ponens holds in this framework.

Having this formula F, we can now compute given say the perspective $w = x_1$ how likely F is:

This could help with generative models, where one would start with $F = x_1$ and keep adding x such that $F' := F \wedge (x_1 \to x)$ has the largest value among the $x \in X$.

Here is an empirical result with interpretation in prosa:

- **Bedrooms → Area → Bathrooms → Floors → YearBuilt → Price → Garage_Yes → Location_Rural → Location_Suburban → Location_Urban → Condition_Fair → Condition_Good**:

  This chain starts from **Bedrooms**, implying a logical sequence of attributes from bedrooms to good condition. This chain holds **True** with a confidence of 0.7227.

- **Bathrooms → Area → Bedrooms → Floors → YearBuilt → Price → Garage_Yes → Location_Rural → Location_Suburban → Location_Urban → Condition_Fair → Condition_Good**:

  Starting from **Bathrooms**, the sequence leads to good condition. This chain holds **True** with a confidence of 0.7143.

- **Floors → Price → Area → Bedrooms → Bathrooms → YearBuilt → Location_Rural → Location_Suburban → Location_Urban → Condition_Fair → Condition_Good → Condition_Poor**:

  Beginning with **Floors**, this chain implies eventual poor condition. This chain holds **True** with a confidence of 0.6920.

- **YearBuilt → Price → Area → Bedrooms → Bathrooms → Floors → Location_Suburban → Location_Rural → Location_Urban → Condition_Fair → Condition_Good → Condition_Poor**:

  Starting from **YearBuilt**, the chain eventually leads to poor condition. This chain holds **True** with a confidence of 0.7523.

- **Price → Area → Bedrooms → Bathrooms → Floors → YearBuilt → Garage_Yes → Location_Rural → Location_Suburban → Location_Urban → Condition_Fair → Condition_Good**:

  Starting from **Price**, the chain leads to good condition. This chain holds **True** with a confidence of 0.7573.

- **Location_Rural**:

  The location is rural, and this property is **True** with a confidence of 1.0000.

- **Location_Suburban**:

  The location is suburban, and this property is **True** with a confidence of 1.0000.

- **Location_Urban**:

    The location is urban, and this property is **True** with a confidence of 1.0000.

- **Condition_Fair → Price → Area → Bedrooms → Bathrooms → Floors → YearBuilt → Location_Rural → Location_Suburban → Location_Urban → Condition_Good → Condition_Poor**:

    Starting from **Condition_Fair**, the sequence indicates a downward progression to poor condition. This chain is **Indeterminate** with a confidence of 0.4649.

- **Condition_Good**:

    The condition of the property is good, and this property is **True** with a confidence of 1.0000.

- **Condition_Poor**:

    The condition of the property is poor, and this property is **True** with a confidence of 1.0000.

11 Conclusion

In this document, we have developed and formalized a semantic space framework that captures logical reasoning in a geometric context using Reproducing Kernel Hilbert Spaces (RKHS). Through this construction, logical operations such as conjunction, disjunction, negation, implication, and quantification have been defined using inner products and projections in the RKHS. The framework adheres to classical logical properties, ensuring compatibility with established logical systems while providing a novel geometric perspective.

Our analysis demonstrates the versatility of the framework in various applications, such as propositional and first-order logic, as well as conceptual spaces. This approach offers advantages in handling uncertainty, ambiguity, and indeterminate truth values, which are particularly useful in fields like artificial intelligence, cognitive science, and knowledge representation. Furthermore, by connecting algebraic and geometric methods, the framework opens new avenues for exploring the logical structure of complex data and conceptual relationships.

Overall, the semantic space framework provides a powerful and intuitive tool for reasoning about logical systems within a geometric paradigm, enhancing both theoretical understanding and practical applications in diverse domains.

12 Appendix: Python Code

In this section we list some Python code to show the application of the Lukasiewicz semantic space of logic for the House Price Prediction Dataset which is on Kaggle.

```python
# Code for generating random hypothesis / propositional formulas and
    testing them:
import random

import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, MinMaxScaler
from sklearn.metrics.pairwise import cosine_similarity

# Logical Operations as per Lukasiewicz Logic
def AND(a, b):
    return np.minimum(a, b)

def OR(a, b):
    return np.maximum(a, b)

def NOT(a):
    return 1 - a

def IMPLIES(a, b):
    return np.minimum(1, 1 - a + b)

def IFF(a, b):
    return 1 - np.abs(a - b)


def existential_quantifier(t_values):
    """
    Existential quantifier Exists x in M, P(x). Takes the maximum value
        from the list of t_values.
    """
    return max(t_values)

# Compute Forall x in M, P(x) (universal quantifier)
def universal_quantifier(t_values):
    """
    Universal quantifier Forall x in M, P(x). Takes the minimum value
        from the list of t_values.
    """
    return min(t_values)

# Evaluation Function
def evaluate(value):
    eps = 5*1e-2
    if value > 0.5+eps:
        return 'True'
    elif 0.5-eps <= value and value <= 0.5+eps:
        return 'Indeterminate'
    else:
        return 'False'

# Load the Bank Marketing Dataset
```

```python
def load_data(filepath,sep=";"):
    df = pd.read_csv(filepath, sep=sep)
    return df

# Preprocess the Data
def preprocess_data(df,remove_columns = []):
    # Separate features
    X = df
    for rm in remove_columns:
        X = X.drop(rm,axis=1)

    # Identify categorical and numerical columns
    categorical_cols = X.select_dtypes(include=['object']).columns
    numerical_cols = X.select_dtypes(include=['int64', 'float64']).columns
    print(categorical_cols,numerical_cols)

    # One-Hot Encode Categorical Variables
    encoder = OneHotEncoder(sparse_output=False, drop='first')
    X_encoded = pd.DataFrame(encoder.fit_transform(X[categorical_cols]),
                             columns=encoder.get_feature_names_out(categorical_cols))

    X_numerical = X[numerical_cols]

    # Scale Numerical Variables
    scaler = MinMaxScaler()
    X_scaled = pd.DataFrame(scaler.fit_transform(X[numerical_cols]),
                            columns=numerical_cols)

    # Combine Encoded and Scaled Features
    X_processed = pd.concat([X_scaled, X_encoded], axis=1)

    X_final = X_processed.dropna()

    return X_final

# Build Similarity Matrix
def build_similarity_matrix(X):
    similarity = cosine_similarity(X.T)
    similarity_df = pd.DataFrame(similarity, index=X.columns, columns=X.columns)
    return similarity_df

def generate_random_formula(variables, gram_matrix, perspective, max_depth=3, current_depth=0):
    """
    Recursively generate a random propositional formula.

    Args:
        variables (list of str): List of propositional variables.
```

```
        max_depth (int): Maximum depth of the formula.
        current_depth (int): Current depth in the recursion.

    Returns:
        str: A random propositional formula as a string.
    """
    df = gram_matrix
    w = perspective
    dd = {"AND":AND,"IFF":IFF,"IMPLIES":IMPLIES,"NOT":NOT,"OR":OR}

    # Base case: return a variable, possibly negated
    if current_depth >= max_depth or (current_depth > 0 and random.
        random() < 0.3):
        var = random.choice(variables)
        proj = df.loc[w,var]
        if random.random() < 0.3:
            return f'(NOT {var})',NOT(proj)
        else:
            return var,proj
    else:
        # Choose a binary operator
        operator = random.choice(['AND', 'OR', 'IMPLIES', 'IFF'])
        opr = dd[operator]

        # Recursively generate left and right sub-formulas
        left = generate_random_formula(variables, gram_matrix,
            perspective, max_depth, current_depth + 1)
        right = generate_random_formula(variables, gram_matrix,
            perspective, max_depth, current_depth + 1)

        return f'({left[0]} {operator} {right[0]})',opr(left[1],right
            [1])

def generate_multiple_formulas(variables, similarity_df, perspective,
    num_formulas=5, max_depth=3):
    """
    Generate multiple random propositional formulas.

    Args:
        variables (list of str): List of propositional variables.
        num_formulas (int): Number of formulas to generate.
        max_depth (int): Maximum depth of each formula.

    Returns:
        list of str: A list containing generated propositional formulas
            .
    """
    formulas = []
    for _ in range(num_formulas):
        formula = generate_random_formula(variables, similarity_df,
            perspective, max_depth)
        formulas.append(formula)
```

```
    return formulas


if __name__ == "__main__":
    filepath = "./kaggle_datasets/House Price Prediction Dataset.csv"
    sepHouse= ","

    sep = sepHouse
    #print(description)

    # Load Data
    print("Loading data...")
    df = load_data(filepath,sep)

    # Preprocess Data
    print("Preprocessing data...")
    X_processed = preprocess_data(df,remove_columns = ["Id"])

    # Build Similarity Matrix
    print("Building similarity matrix...")
    similarity_df = build_similarity_matrix(X_processed)
    print(similarity_df.columns)

    # Define the list of variables
    X = similarity_df.columns.tolist()


    for perspective in X:
        # Generate 5 random propositional formulas

        formulas_found = 0
        while not formulas_found>5:
            random_formulas = generate_multiple_formulas(X, similarity_df, perspective, num_formulas=150, max_depth=1)
            evs = []
            # Display the generated formulas

            for idx, formula in enumerate(random_formulas, 1):
                if formula[1]>0.45 and formula[1]<0.55 or formula[1]<0.15 or formula[1]>0.85:
                    evs.append((formula[1],evaluate(formula[1]),formula))
                    formulas_found += 1
                    print("Randomly Generated Propositional Formulas for perspective (",perspective,"):")
                    print(evs[-1])
                    break
```

```
# Code for generating sequence of likely implications starting from a property / desired perspective:
import random
```

```python

import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, MinMaxScaler
from sklearn.metrics.pairwise import cosine_similarity

# Logical Operations as per Lukasiewicz Logic
def AND(a, b):
    return np.minimum(a, b)

def OR(a, b):
    return np.maximum(a, b)

def NOT(a):
    return 1 - a

def IMPLIES(a, b):
    return np.minimum(1, 1 - a + b)

def IFF(a, b):
    return 1 - np.abs(a - b)



# Evaluation Function
def evaluate(value):
    eps = 5*1e-2
    if value > 0.5+eps:
        return 'True'
    elif 0.5-eps <= value and value <= 0.5+eps:
        return 'Indeterminate'
    else:
        return 'False'

# Load the Bank Marketing Dataset
def load_data(filepath,sep=";"):
    df = pd.read_csv(filepath, sep=sep)
    return df

# Preprocess the Data
def preprocess_data(df,remove_columns = []):
    # Separate features
    X = df
    for rm in remove_columns:
        X = X.drop(rm,axis=1)

    # Identify categorical and numerical columns
    categorical_cols = X.select_dtypes(include=['object']).columns
    numerical_cols = X.select_dtypes(include=['int64', 'float64']).columns
    print(categorical_cols,numerical_cols)

```

```
    # One-Hot Encode Categorical Variables
    encoder = OneHotEncoder(sparse_output=False, drop='first')
    X_encoded = pd.DataFrame(encoder.fit_transform(X[categorical_cols]),
                             columns=encoder.get_feature_names_out(categorical_cols))

    X_numerical = X[numerical_cols]

    # Scale Numerical Variables
    scaler = MinMaxScaler()
    X_scaled = pd.DataFrame(scaler.fit_transform(X[numerical_cols]),
                            columns=numerical_cols)

    # Combine Encoded and Scaled Features
    X_processed = pd.concat([X_scaled, X_encoded], axis=1)

    X_final = X_processed.dropna()

    return X_final


def kk(a,b,df=similarity_df):
    return df.loc[a,b]

def proj(kk,a,b):
    return kk(a,b,df=similarity_df)

def test_sequence(kk,seq):
    # linear logic
    #print(f"\nTesting {description} sequence: {' -> '.join(txt)}")
    w = seq[0]
    start = proj(kk, w, w)  # Start with the assumption that the first word's projection is true

    for i in range(len(seq) - 1):
        xi = seq[i]
        pxi = proj(kk, w, xi)
        xj = seq[i+1]
        pxj = proj(kk,w,xj)
        impl = IMPLIES(pxi,pxj)
        start = AND(start, impl)
    truth_val = evaluate(start)
    #print(seq,truth_val)
    #print(f"Sequence is evaluated as: {truth_val} ({start:.3f})")
    return (truth_val,start)

# Evaluate both sequences


def generate_next_token(kk,ll,X):
    max_prob = -2
```

```
    max_word = ""
    for w in X:
        seqs = l1+[w]
        tv,prob = test_sequence(kk,seqs)
        #print(seqs,tv,prob)
        if prob > max_prob and not (w in  l1):
            max_word = w
            max_prob = prob
    if max_word!="":
        return l1+[max_word]
    else:
        return l1
def generate_meaningful_sequence(kk,l1,Ngen,X,verbose=False):
    if Ngen is None:
        l10 = [l for l in l1]
        while test_sequence(kk,l10)[0]!="False":
            l11 = generate_next_token(kk,l10,X)
            if verbose:
                print(l10,test_sequence(kk,l10))
            if len(l10)==len(l11):
                break
                return l10
            else:
                l10 = [l for l in l11]
        l10.pop(-1)
        return l10
    else:
        for k in range(Ngen):
            l1 = generate_next_token(kk,l1,X)
        return l1


filepath = "./kaggle_datasets/House Price Prediction Dataset.csv"
sepHouse= ","

sep = sepHouse
    #print(description)

    # Load Data
print("Loading data...")
df = load_data(filepath,sep)

    # Preprocess Data
print("Preprocessing data...")
X_processed = preprocess_data(df,remove_columns = ["Id"])

    # Build Similarity Matrix
print("Building similarity matrix...")
similarity_df = build_similarity_matrix(X_processed)
print(similarity_df.columns)

    # Define the list of variables
```

```python
X = similarity_df.columns.tolist()
for x in X:
    seq = (generate_meaningful_sequence(kk,ll=[x],Ngen=None,X=X))
    print(seq)
    print(test_sequence(kk,seq))
```

```python
# Data output for the sequences
['Area', 'Price', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Garage_Yes', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good']
('True', 0.7573235376101951)
['Bedrooms', 'Area', 'Bathrooms', 'Floors', 'YearBuilt', 'Price', 'Garage_Yes', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good']
('True', 0.7226975545155016)
['Bathrooms', 'Area', 'Bedrooms', 'Floors', 'YearBuilt', 'Price', 'Garage_Yes', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good']
('True', 0.7142987211615484)
['Floors', 'Price', 'Area', 'Bedrooms', 'Bathrooms', 'YearBuilt', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good', 'Condition_Poor']
('True', 0.6920406748066003)
['YearBuilt', 'Price', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'Location_Suburban', 'Location_Rural', 'Location_Urban', 'Condition_Fair', 'Condition_Good', 'Condition_Poor']
('True', 0.7523090247901313)
['Price', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Garage_Yes', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good']
('True', 0.7573235376101957)
['Location_Rural']
('True', 1.0000000000000007)
['Location_Suburban']
('True', 1.0)
['Location_Urban']
('True', 1.000000000000001)
['Condition_Fair', 'Price', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Good', 'Condition_Poor']
('Indeterminate', 0.4649147843022506)
['Condition_Good']
('True', 1.0000000000000007)
['Condition_Poor']
('True', 0.9999999999999998)
['Garage_Yes', 'Price', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Location_Rural', 'Location_Suburban', 'Location_Urban', 'Condition_Fair', 'Condition_Good']
('True', 0.6043995507236861)
```

```python
# Data output for the random formulas generated and evaluated:
Randomly Generated Propositional Formulas for perspective ( Area ):
```

```
(1.0, 'True', ('(Condition_Good IMPLIES Bathrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Area ):
(0.8572502552118751, 'True', ('((NOT Bathrooms) IFF Location_Urban)', 0.8572502552118751))
Randomly Generated Propositional Formulas for perspective ( Area ):
(1.0, 'True', ('(Bathrooms IMPLIES Bedrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Area ):
(1.0, 'True', ('(Condition_Poor IMPLIES Floors)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Area ):
(0.9687871220777605, 'True', ('((NOT Location_Suburban) IFF Garage_Yes)', 0.9687871220777605))
Randomly Generated Propositional Formulas for perspective ( Area ):
(0.8703181543645824, 'True', ('(Condition_Fair IFF (NOT Floors))', 0.8703181543645824))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(0.9431369688026381, 'True', ('((NOT Location_Suburban) IFF Garage_Yes)', 0.9431369688026381))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(0.45214452201066824, 'Indeterminate', ('((NOT Garage_Yes) AND Area)', 0.45214452201066824))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(1.0, 'True', ('(Location_Urban IMPLIES Bathrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(-8.881784197001252e-16, 'False', ('(Bedrooms IMPLIES (NOT Bedrooms))', -8.881784197001252e-16))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(0.5478554779893318, 'Indeterminate', ('(Area AND Garage_Yes)', 0.5478554779893318))
Randomly Generated Propositional Formulas for perspective ( Bedrooms ):
(1.0, 'True', ('(Location_Rural IMPLIES Floors)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(0.9418456790147741, 'True', ('(Floors IMPLIES (NOT Condition_Poor))', 0.9418456790147741))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(1.0, 'True', ('(Location_Urban IMPLIES Location_Suburban)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(0.9830213635688818, 'True', ('(Location_Suburban IFF Condition_Poor)', 0.9830213635688818))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(1.0, 'True', ('((NOT Location_Rural) IFF (NOT Location_Rural))', 1.0))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(0.9631318469890159, 'True', ('((NOT Location_Urban) IMPLIES Garage_Yes)', 0.9631318469890159))
Randomly Generated Propositional Formulas for perspective ( Bathrooms ):
(0.8637774755162289, 'True', ('(Garage_Yes IFF YearBuilt)', 0.8637774755162289))
```

```
Randomly Generated Propositional Formulas for perspective ( Floors ):
(0.9198556751629549, 'True', ('(Condition_Poor IFF (NOT Area))',
    0.9198556751629549))
Randomly Generated Propositional Formulas for perspective ( Floors ):
(1.0, 'True', ('((NOT Floors) IMPLIES (NOT Bedrooms))', 1.0))
Randomly Generated Propositional Formulas for perspective ( Floors ):
(1.0, 'True', ('(Condition_Good IMPLIES YearBuilt)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Floors ):
(0.9615264095041067, 'True', ('(Area IFF Bathrooms)',
    0.9615264095041067))
Randomly Generated Propositional Formulas for perspective ( Floors ):
(0.8939740155548392, 'True', ('(Garage_Yes IFF Bathrooms)',
    0.8939740155548392))
Randomly Generated Propositional Formulas for perspective ( Floors ):
(1.0, 'True', ('(Condition_Good IMPLIES YearBuilt)', 1.0))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(0.9650762201722981, 'True', ('(Bathrooms IFF Floors)',
    0.9650762201722981))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(1.0, 'True', ('((NOT Bedrooms) IMPLIES Bathrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(0.968240495721197, 'True', ('(Condition_Poor IMPLIES Location_Urban)',
     0.968240495721197))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(0.9999999999999996, 'True', ('(Condition_Good OR YearBuilt)',
    0.9999999999999996))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(0.9798886955519408, 'True', ('(Condition_Fair IFF Condition_Poor)',
    0.9798886955519408))
Randomly Generated Propositional Formulas for perspective ( YearBuilt )
    :
(1.0, 'True', ('(Bathrooms IMPLIES Area)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Price ):
(0.4649147843022498, 'Indeterminate', ('(Location_Rural OR
    Condition_Fair)', 0.4649147843022498))
Randomly Generated Propositional Formulas for perspective ( Price ):
(1.0, 'True', ('(Location_Suburban IMPLIES (NOT Condition_Poor))', 1.0)
    )
Randomly Generated Propositional Formulas for perspective ( Price ):
(0.9999999999999997, 'True', ('(Bathrooms OR Price)',
    0.9999999999999997))
Randomly Generated Propositional Formulas for perspective ( Price ):
(1.0, 'True', ('((NOT Area) IMPLIES Bathrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Price ):
(0.9397316500164674, 'True', ('(YearBuilt IMPLIES Floors)',
    0.9397316500164674))
Randomly Generated Propositional Formulas for perspective ( Price ):
```

```
(0.9557981902429034, 'True', ('(YearBuilt IMPLIES Bedrooms)',
    0.9557981902429034))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(0.862860575548185, 'True', ('(Bathrooms IFF Condition_Poor)',
    0.862860575548185))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(0.9998553190866013, 'True', ('(YearBuilt IMPLIES Area)',
    0.9998553190866013))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(1.0, 'True', ('((NOT Bathrooms) IMPLIES Location_Rural)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(1.0, 'True', ('(Condition_Good IMPLIES (NOT Condition_Fair))', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(1.0, 'True', ('(Garage_Yes IFF Garage_Yes)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Rural ):
(0.0, 'False', ('(Location_Urban AND Location_Urban)', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(0.982063933440659, 'True', ('((NOT Condition_Fair) IFF (NOT
    Condition_Good))', 0.982063933440659))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(1.0, 'True', ('(Bedrooms IMPLIES Bathrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(1.0, 'True', ('(Location_Rural IMPLIES YearBuilt)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(1.0, 'True', ('(Location_Rural IMPLIES Location_Rural)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(0.0, 'False', ('(Bedrooms AND (NOT Location_Suburban))', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Suburban ):
(0.0, 'False', ('((NOT Area) AND Location_Rural)', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Location_Urban ):
(0.8516301960509322, 'True', ('(Bathrooms IMPLIES Condition_Fair)',
    0.8516301960509322))
Randomly Generated Propositional Formulas for perspective (
    Location_Urban ):
(1.000000000000001, 'True', ('((NOT Location_Rural) OR Location_Urban)'
    , 1.000000000000001))
Randomly Generated Propositional Formulas for perspective (
    Location_Urban ):
(1.0, 'True', ('((NOT Bathrooms) IMPLIES Location_Urban)', 1.0))
```

```
Randomly Generated Propositional Formulas for perspective (Location_Urban ):
(0.8651232306429956, 'True', ('(Bathrooms IMPLIES Condition_Poor)', 0.8651232306429956))
Randomly Generated Propositional Formulas for perspective (Location_Urban ):
(1.0, 'True', ('(Location_Suburban IMPLIES Garage_Yes)', 1.0))
Randomly Generated Propositional Formulas for perspective (Location_Urban ):
(1.0000000000000001, 'True', ('(Location_Urban OR (NOT Condition_Good))', 1.0000000000000001))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(1.0, 'True', ('(Location_Urban IMPLIES Condition_Fair)', 1.0))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(0.0, 'False', ('(Condition_Good AND (NOT Bathrooms))', 0.0))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(1.0, 'True', ('(Location_Rural OR (NOT Condition_Poor))', 1.0))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(0.8703145558524861, 'True', ('(Location_Rural IFF Floors)', 0.8703145558524861))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(0.8821726119294495, 'True', ('((NOT Bedrooms) IFF Price)', 0.8821726119294495))
Randomly Generated Propositional Formulas for perspective (Condition_Fair ):
(0.0, 'False', ('(Condition_Poor AND YearBuilt)', 0.0))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(1.0, 'True', ('(Location_Rural OR (NOT Condition_Poor))', 1.0))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(0.9999999999999993, 'True', ('(Condition_Poor IFF (NOT Condition_Good))', 0.9999999999999993))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(0.0, 'False', ('(Condition_Poor OR Condition_Poor)', 0.0))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(0.9938416973972002, 'True', ('(YearBuilt IFF Area)', 0.9938416973972002))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(0.0, 'False', ('((NOT Location_Suburban) AND Condition_Fair)', 0.0))
Randomly Generated Propositional Formulas for perspective (Condition_Good ):
(0.8621320644884826, 'True', ('(Floors IFF Location_Suburban)', 0.8621320644884826))
```

```
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(0.8622070352025419, 'True', ('(Location_Urban IFF Floors)',
    0.8622070352025419))
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(0.0, 'False', ('(Location_Suburban AND Condition_Good)', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(1.0, 'True', ('(Location_Rural IMPLIES Location_Rural)', 1.0))
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(0.0, 'False', ('(Condition_Good AND (NOT Area))', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(0.0, 'False', ('(Condition_Fair AND (NOT Garage_Yes))', 0.0))
Randomly Generated Propositional Formulas for perspective (
    Condition_Poor ):
(1.0, 'True', ('(Location_Urban IFF Location_Urban)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(0.994771274954062, 'True', ('((NOT Condition_Good) IMPLIES (NOT
    Condition_Poor))', 0.994771274954062))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(0.45214452201066824, 'Indeterminate', ('(Bedrooms IFF (NOT Garage_Yes)
    )', 0.45214452201066824))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(1.0, 'True', ('(Condition_Fair IFF Condition_Fair)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(1.0, 'True', ('(Location_Urban IMPLIES Bedrooms)', 1.0))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(0.9042890440213365, 'True', ('(Bedrooms IFF (NOT Bedrooms))',
    0.9042890440213365))
Randomly Generated Propositional Formulas for perspective ( Garage_Yes
    ):
(0.5356905325519948, 'Indeterminate', ('(Floors AND Area)',
    0.5356905325519948))
```

Literatur

[1] Kaggle, "House Price Prediction Dataset," Kaggle, 2024. Available: https://www.kaggle.com/datasets/zafarali27/house-price-prediction-dataset/data.

[2] Antti Hautamäki, *A Perspectivist Approach to Conceptual Spaces*, Synthese, 2016.