Conceptual Topology × Category Theory As a Semantics Cage: Toward an Algebraic Topology of Meaning

Written by No Name Yet Exist
Contact: Written Below

Introduction

We introduce a formal framework for Semantic Algebraic Operators, as an extension of the Conceptual Topology approach to semantic representation. The framework aims to model how meaning can be composed, transformed, and analyzed through algebraic operations on morphic flow structures, providing a foundation for the development of Semantic Operator Algebra. In this work, we formalize key operations such as semantic shifting (σ), semantic morphism set operators (⊕, ⊖), and cross-category flow conditions. Building on the Monoid structure of morphic flows (M|Z), the framework captures both safe and rupture-inducing transformations of meaning. This establishes a foundation for future developments in Semantic Operator Algebra, with potential applications to semantic DSLs, cognitive modeling, and cross-linguistic analysis.

Index

Global Definition of Conceptual Topology: Morphic Chain Structure

```
CT := (C, B, π: E → B, Fb := π⁻¹(b), A ≅ b ⋃ Nat(Hom(−, A), Fb))

Where:
  - C is the category of concepts (objects = words or concepts)
  - B is the base space of Z-frames (semantic continuity anchors)
  - E is the total semantic space (word vector embedding space)
  - π projects each concept to its semantic base (Z-frame)
  - Fb is the fiber (semantic morphic chain) over a base b
  - A ≅ b ⋃ Nat(Hom(−, A), Fb) interprets each concept A via its morphisms
relative to its   Z-frame b (Yoneda perspective defined in appendix)
```

```
        Z = baby
    puppy  ←   girl    //specified: size + young
     |          |
     |          |
    dog        she     //abstraction
      \        /
        Mammal
```

Local Semantic Flow under Z Frame

Identity Morphism

In Category Theory, Identity Morphism is always defined.

```
id_X: X → X
such that for any f: X → Y:
f ∘ id_X = f
id_Y ∘ f = f
```

However, in Conceptual Topology, morphisms are mediated by Z frame, thus the identity morphis is not always given unless Z is defined.

Two Types of Identity Morphism in Conceptual Topology

1. f: X → X | X (Category-theoretic identity)

```
f: X → X | X
such that for any f: X → Y:
f ∘ id_X = f
id_Y ∘ f = f

e.g. f: dog → dog | dog
```

2. f: X → X | Z (Mediated identity with semantic flow)

```
f: X → X | Z

f: X → Z
f⁻¹: Z → X
f⁻¹ ∘ f ≅ id_X
e.g. you → you | externalized perspective
     NL: you are you
```

Since the identity morphism passes through an external anchor point, the identity morphism is defined quasi-identical.
e.g. dog → perro | собака

Simplified Form of Identity Morphism:

1. f: X → X | X (Category-theoretic identity)
   In simplified form: X
   or more explicitly: id_X

2. f: X → X | Z (Mediated identity with semantic flow)
   In simplified form: X | Z

Mirror Morphism Definition:

Each mirror maps conceptual transitions across vocabularies while preserving morphic identity up to rupture—that is, it allows for semantic divergence that still respects underlying structural continuity, even if exact invertibility is not preserved.

```
f : X  → Y  | Z ∈ Dᵢ
f': X' → Y' | Z ∈ Dᵢ₊₁
  ⇒X' ≠ X, but cod(f) = cod(f') | CD (CD = codomain)

We define f' as a mirror-correspondent morphism of f under a given Z-frame,
if and only if:

∃Z: rupture(f, f' | Z) ≠ ∅
∧ cod(f) = cod(f') | CD
```

```
        Z = baby
    puppy  ←   girl    //specified: size + young
     |          |
     |          |
    dog        she     //abstraction
      \        /
        Mammal
```

Note: Z: rupture(f, f' | Z) ≠ ∅ means that there exists a Z-frame under which *f* and *f'* exhibit structural divergence—i.e., they are not fully invertible but still converge at the codomain level.

For example, let *Z = abstraction*. This allows a semantic transition from *girl* → *she* and *puppy* → *dog*, treating them as mirror morphisms under a shared conceptual frame.
However, if we take *Z = agency*, a rupture emerges: *puppy* → *dog* lacks agency, while *girl* → *she* retains it.
Hence, **rupture(f, f' | agency) ≠ ∅**, yet *f* and *f'* still align toward the same codomain (e.g., *mammal*).

Quasi-Natural Transformation of Meaning Systems

A **Morphic Chain Mirror** is a contextual correspondence between two morphic chains drawn from distinct but meaning-aligned vocabularies. This correspondence is realized through a **quasi-natural transformation** under a shared intermidiating Z-frame.

```
η: Dᵢ ⇒ Dᵢ₊₁  | CD (CD = codomain)
η_X ∘ Dᵢ({f₁ | Z₁, ..., fₙ | Zₙ}) ≈ Dᵢ₊₁({f'₁ | Z₁, ..., f'ₙ | Zₙ}) ∘ η_Y | CD
for all fⱼ: Xⱼ → Yⱼ | Zⱼ ∈ Dᵢ,
where f'ⱼ: η_X(Xⱼ) → η_Y(Yⱼ) | Zⱼ

Then, η is said to be a quasi-natural transformation under the Z-frame
i.e. η ∈ Mor(C) where C is the contextual meaning category
Example: η: girl → puppy | Z = Baby
```

Monoid Structure of Semantic Flow (M|Z)

In Conceptual Topology, Z is defined as a mediating point/semantic anchor.

```
Let C and D, Z be categories,
with semantic projection π: C ∪ D → Z, such that for each X ∈ Ob(C ∪ D):

π(X) ∈ Ob(Z)

For each X ∈ Ob(C ∪ D), there exists morphism:

f_X: X → π(X)
f_X⁻¹: π(X) → X

such that:

f_X⁻¹ ∘ f_X ≅ id_X

For morphism f: X → Y | Z,
this corresponds to:

f_Z: π(X) → π(Y) in Z

For any X, Y ∈ Ob(C ∪ D):

Let [X]_Z := semantic representation of X under frame Z (i.e., π(X))

Then:

[ X ]_Z1 ≅ [ Y ]_Z2 | Z1, Z2 ∈ Z //or Z1, Z2 >> Z
which means:
["Dog"]_Pet = [Retriever, Dachshund, Poodle, Bulldog, ...]
["girl"]_Human = [girl, woman, person, ...]
["Dog"]_Pet ≅ ["girl"]_Human | Life


Then the set of semantic flow morphisms under Z forms a monoid:

M|Z = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z }
```

Example: Queen and King

```
Let C be a concept
C = {-OC₁, -OC₂, ..., -OCₙ} // Yoneda's lemma and structralism

King = {Royalty□, Male□, Human□}
Queen = {Royalty□, Female□, Human□}

k₁: human → male | King
k₂: male → royalty | King

Then:
k₂ ∘ k₁: human → royalty | King

q₁: human → female | Queen
q₂: female → royalty | Queen

Then:
q₂ ∘ q₁: human → royalty | Queen

Cross-Z-frame composition:

(k₂ ∘ k₁) ⊕ (q₂ ∘ q₁)= Human → Royalty | Z'

If higher Z' (e.g. "Humanity" or "Social Role") unifies Z = King and Z =
Queen:

Then:
(k₂ ∘ k₁) ⊕ (q₂ ∘ q₁) ∈ M|Z'
```

The $\oplus$ operator enables morphic chains to merge while preserving Monoid coherence under Z', resulting in $(k_2 \circ k_1) \oplus (q_2 \circ q_1) \in M|Z'$.

Identity Element of M|Z

```
Let: M|Z = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z }

Define the identity element of M|Z as a family of identity morphisms over
the shared Z frame:

For each X ∈ Ob(C ∪ D), there exists a unique identity morphism under a Z
frame:

e|Z_X := id_X | Z

Then, for any f: X → Y | Z ∈ M|Z:

e|Z_X ∘ f = f
f ∘ e|Z_Y = f

Therefore, the identity structure of M|Z is given by the family:

{ id_X | Z | X ∈ Ob(C ∪ D) }

which forms a pointwise identity across the objects under the common Z
frame.
This ensures that M|Z satisfies the identity axiom of a monoid.
```

Associativity of M|Z

```
Let: M|Z = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z }

Then for all f, g, h ∈ M|Z:

(f ∘ g) ∘ h = f ∘ (g ∘ h)

Thus, the composition ∘ in M|Z is associative.
```

Example:

```
σ_id(Word). Op(word, Word) = word → σ_id = e|Z
Note: The definition of σ is provided in the following section.
```

Example: Cross Category Identity Morphism

```
dog ∈ Ob(Pet)
dog ∈ Ob(Animal)

dog → dog | Z=Domesticated
```

Definition: Semantic Shifting Morphism (σ)

```
σ: D(X_{n-1} | X) → D(X_{n-1} | X)

such that σ ⊕ f ∈ M|Z if and only if type compatibility holds:

∀ A, B, (A → B) ∘ σ(X) is valid if:

( A >> X or X >> A )
and
( B >> X or X >> B )

Definition: Subsumption
A >> X ≡ A ⊑ X

Definition: SubsumedBy
X >> A ≡ X ⊑ A

Example:
king → king >> human → human
  ⇒ king >> human  → valid

human → human >> queen → queen
  ⇒ human >> queen → valid
```

Semantic Operators Semantic Operator σ modifies morphism as follows.

```
σ(X). Not(x){ A ↛ B | Z}     →     Rupture under Z frame
σ(X). so_much(x){A → B | Z}  →     Preservation & amplification under Z frame
σ(X). >>(x,y)                →     Semantic Shifting x to y (Generalization) as function form
σ(X). <<(x,y)                →     Downward Shifting x to y (Specialization) as function form
σ(X). >(x,y)                 →     Semantic Shifting
```

Semantic Morphism Set Operators

```
Addition (⊕):
σ(X). ⊕(f, Aₙ₋₁ | Z): D(Aₙ₋₁ | Z) → D(Bₙ₋₁ | Z) | Z
σ(X). ⊕(f₁, f₂) : Aₙ₋₁ :={f₁, f₂}

Subtraction (⊖):
⊖: Aₙ₋₁ ⊖ {f_i}
σ(X). ⊖(f, Aₙ₋₁ | Z): D(Aₙ₋₁ | Z) → D(Bₙ₋₁ | Z) | Z

- ⊕ operator is σ_safe if Z alignment is preserved.
- ⊖ operator is potentially σ_unsafe but can be σ_safe if resulting chain
preserves the underlying morphic continuity Z.
```

Example

```
{Royalty▯, Male▯, Human▯} ⊖ {Male▯} ⊕ {Female▯}
= {Royalty▯, Female▯, Human▯}
= queen
```

Semantic Shifting

```
C_chain = { f₁, f₂, ..., fₙ | Z} ∈ D(Cₙ₋₁ | Z)

σ(X): D(Aₙ₋₁ | Z) > D(Bₙ₋₁ | Y) | Z, Y ∈ CD
where:
D(Aₙ₋₁ | Z) = source morphic chain
D(Bₙ₋₁ | Y) = target morphic chain
CD = codomain alignment (semantic anchor)

σ(X) is not strict functorial → quasi-alignment under semantic equivalence

σ(X) ≈ η: Dᵢ ⇒ Dᵢ₊₁ | CD (Quasi-Natural Transformation interpretation)

Example:
σ(X). >(puppy → dog → mammal | Canine, Human) ∋ girl → she → mammal |
Human

where: canine, Human ∈ Mammal
```

Identity Morphism of σ

```
word is word
thus:
word ≅ Nat(Hom(-, word), Fib(word))

σ_id(Z). OP(X,Z) = σ such that σ(f) = f for all f ∈ Hom(X, X) unless OP is
σ_unsafe such that word is not a word: σ(Word). Not(word ↛ word)

σ_id(Word). OP(word, Word) = word

σ_id(Word). OP(f, Word) = f  for all f: word → word | word
since: M|Z = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z
= Word }

σ_id ∈ M|Z
σ ∘ σ_id = σ
σ_id ∘ σ = σ
∴ word is word and word is word
```

Associativity of σ

```
σ₁(Z). OP(D(Aₙ₋₁ | Z), Z) = D(Zₙ₋₁ | Z)
σ₂(Z). OP(D(Bₙ₋₁ | Z), Z) = D(Zₙ₋₁ | Z)

Then the composition σ₂ ∘ σ₁:
σ_comp(Z). OP(D(Zₙ₋₁ | Z), D(Zₙ₋₁ | Z)) = D(Zₙ₋₁ | Z)

where: OP is not σ_unsafe and under shared Z frame

Associativity
For all σ₁, σ₂, σ₃ such that their domains/codomains match for composition:
(σ₃ ∘ σ₂) ∘ σ₁ = σ₃ ∘ (σ₂ ∘ σ₁)

Thus, σ composition operator is associative under Monoid structure.
```

Example:

```
Let σ₁ = σ(Mammal). >>(canine → mammal, Life)  = (life → life | Life)
Let σ₂ = σ(Mammal). >>(mammal → animal, Life)  = (life → life | Life)
Let σ₃ = σ(Mammal). >>(animal → livingBeing, Life) = (life → life | Life)

Conclusion:
 (σ₃ ∘ σ₂) ∘ σ₁ = σ₃ ∘ (σ₂ ∘ σ₁) =  (life → life | Life)
```

x Safe / Unsafe Semantic Shifting Morphism (σ)

Definition of Safe and Unsafe σ Operator
Semantic Shifting Morphism (σ) can be classified based on whether it preserves the global coherence of the morphic chain.

Safe σ Operator (σ_safe) Acts on the entire morphic chain as a coherent transformation.

```
σ_safe: D(Aₙ₋₁ | Z) > D(Bₙ₋₁ | Z') | Z >> Z' ∨ Z << Z'

where: Z, Z'∈ CD
```

Behaves as a Quasi-Natural Transformation

```
σ_safe ≈ η: Dᵢ ⇒ Dᵢ₊₁ | CD
```

Composition is associative:

```
(σ₃ ∘ σ₂) ∘ σ₁ = σ₃ ∘ (σ₂ ∘ σ₁)
```

Resulting chain remains in M|Z or M_{Z'} (closed).

Example

```
σ₁(X). >(canine, mammal)
σ₂(X). >(mammal, animal)
σ₃(X). >(animal, livingBeing)

Composition:
(σ₃ ∘ σ₂) ∘ σ₁ = σ₃ ∘ (σ₂ ∘ σ₁)
→ >(canine, livingBeing)

Entire morphic chain is preserved.
```

Unsafe σ Operator (σ_unsafe) Does not preserve global coherence of the morphic chain. Acts locally or in a decomposed manner.

Chain may collapse:

```
σ_unsafe: D(Aₙ₋₁ | Z) → { rupture(f₁), rupture(f₂), ..., rupture(fₙ) | ¬Z }
```

```
rupture(f, σ(f) | Z) ≠ ∅
```

Cannot be captured by a Quasi-Natural Transformation globally.

Example

```
σ(X). Not(x) { A ⊣̸ B | Z }

Result:
rupture(A ⊣̸ B | Z)
→ breaks the morphic flow → chain decomposes.
```

Example: Morphic Shifting and Cross-Z-frame Semantic Flow

We consider the well-known semantic analogy: **king - man + woman = queen**

We formalize this using Conceptual Topology's morphic flow.

```
Concept Definitions

Let C be a concept:

// (Yoneda's lemma and structuralism — concept defined by oppositional
components)
C = {−OC₁, −OC₂, ..., −OCₙ}

King = {Royalty□, Male□, Human□}
Queen = {Royalty□, Female□, Human□}

We define king as:

k₁: human → male | King
k₂: male → royalty | King

Then:
k₂ ∘ k₁: human → royalty | King

We define queen as:
```

```
q₁: human → female | Queen
q₂: female → royalty | Queen

Then:
q₂ ∘ q₁: human → royalty | Queen

m⁻¹: male → human | Human
f: human → female | Human

We define semantic editing morphisms σ within the Human frame:

σ(Human). >(m⁻¹ ∘ f, Human)
= (male → human) ∘ (human → female) | Human ⊕ (human → human | Human)
= human → human | Human

since:
M|Human = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z =
Human }

thus:
((male → human | Human) ∘ (human → female | Human)) ∈ M|Human
where: (male → human | Human) ≅ (human → female | Human) under shared
morphic continuity Z
and:
((male → human | Human) ∘ (human → female | Human )) >> human → human |
Human

Thus formally:
σ(Human). >(m⁻¹ ∘ f, Human)
= (male → human)∘(human → female) | Human ⊕ (human → human | Human)
= human → human | Human

Then:
(k₂ ∘ k₁) ∘ σ(Human). >(m⁻¹ ∘ f): Human → Royalty | Human

Assuming a higher semantic frame Z' (e.g., Humanity, Social Role) that
unifies king and queen frames:

(k₂ ∘ k₁) ∘ σ(Human). >(m⁻¹ ∘ f) ∈ M|Z'

Thus:

king - man + woman = queen
is formalized as:

// Represents: shifting Gender component in Human → mapping to Royalty
(k₂ ∘ k₁) ∘ σ(Human). >(m⁻¹ ∘ f) ∈ M|Z'
```

Cross-Category Flow Condition

Conceptual Topology allows communicative diagram across categories as long as same Z is shared: shared semantic equivalence.

```
Let C and D be categories with Z-frame semantic anchoring.
If all morphisms involved are of the form:
f: X → Y | Z

and semantic equivalence holds:
∀ X, Y ∈ Ob(C ∪ D), [X]_Z ≅ [Y]_Z

then semantic flow forms a monoid over Z-frame:
M|Z = { fₙ ∘ ... ∘ f₁ | all fᵢ: Xᵢ → Xᵢ₊₁ | Z ∧ ∀ i, j: fᵢ ≅ fⱼ | Z }
```

Example: Cross-Category Flow under Z-frame
This example illustrates how morphic flow can traverse categories (livestock, pet)
as long as semantic equivalence under Z = domesticated is preserved.

```
(dog ≅ cat ≅ pig) | Z = Domesticated
Ob(livestock) = {pig}
Ob(pet) = {dog, cat}

F₁: pig → cat | Z
f₂: cat → dog | Z
F₃: dog → pig | Z
Then:
F₃ ∘ f₂ ∘ F₁: pig → pig | Z ∈ M|Z
```

Conclusion

In this work, we have proposed a formal framework for Semantic Algebraic Operators, extending the structure of Conceptual Topology and Semantic Circulation. By introducing operators such as σ, ⊕, and ⊖, and formalizing their algebraic properties, we have shown that semantic flow can be systematically represented and manipulated. Future work will explore the development of a complete Semantic Operator Algebra and its applications in semantic DSLs, cognitive modeling, and cross-linguistic analysis.

Future Work

- Full formalization of Semantic Morphism Set Algebra.
- Extension of σ operator to modal and aspectual semantics.
- Empirical validation using vector-based semantic spaces.
- Application to cross-linguistic semantic flow modeling.

Appendix

simbols

```
Z : Intermediating variable (semantic anchor; Z-frame)
| : Frame separator (indicates morphism is mediated by Z-frame)
→: Morphic Flow
↛: Ruptured morphism
F : Cross-category morphism (used in cross-category flow under shared Z-frame)
//: Used to narrate meaning flow of morphic chains.
¬: Absence

M|Z : Monoid of Semantic Flow under Z-frame
R|Z := { rupture(f) | rupture(f, σ(f) | Z) ≠ ∅ }
e|Z : Identity element of M|Z
D(A_{n-1} | Z) : Morphic chain under Z frame

σ : Semantic Shifting Morphism
>> : Generalization relation (A >> X ≡ A ⊑ X)
<< : Specialization relation (X >> A ≡ X ⊑ A)
rupture(f, σ(f) | Z) ≠ ∅ : Indicates semantic rupture
η : Quasi-Natural Transformation: Contextual alignment between morphic chains.

⊕:  Semantic morphism set addition in σ or morphic merger such as:
    (k_2 ∘ k_1) ⊕ (q_2 ∘ q_1) = human → royalty | Z'
⊖:  Semantic morphism set subtraction
    Removes specified morphisms from a morphic chain or set.
```

Notations

```
Concept / Word (lexeme):
    - Lower case (e.g., puppy, dog, girl, she)

Z Frame (semantic anchor):
    - Upper case (e.g., Mammal, Human, Agency, Domesticated, Royalty)

Type variables (A, B, X, Y, Z in formal definitions):
    - Follow standard formal notation (uppercase)

Example:
puppy → dog | Mammal
A → B | Z

Morphism: f, g, h
Functor: F
```

Simplified Form of Identity Morphism:

1. f: X → X | X (Category-theoretic identity)
   In simplified form: X
   or more explicitly: id_X

2. f: X → X | Z (Mediated identity with semantic flow)
   In simplified form: X | Z

σ Operator

```
σ(X). Not(x){ A ↛ B | Z}     →     Rupture under Z frame
σ(X). so_much(x){A → B | Z}  →     Preservation & amplification under Z
frame
σ(X). >>(x,y)                →     Semantic Shifting x to y
(Generalization) as function form
σ(X). <<(x,y)                →     Downward Shifting x to y
(Specialization) as function form
σ(X). >(x,y)                 →     Semantic Shifting
```

Semantic Morphism Set Operators

```
Addition (⊕):
σ(X). ⊕(f, Aₙ₋₁ | Z): D(Aₙ₋₁ | Z) → D(Bₙ₋₁ | Z) | Z
σ(X). ⊕(f₁, f₂) : Aₙ₋₁ :={f₁, f₂}

Subtraction (⊖):
⊖: Aₙ₋₁ ⊖ {f_i}
σ(X). ⊖(f, Aₙ₋₁ | Z): D(Aₙ₋₁ | Z) → D(Bₙ₋₁ | Z) | Z

- ⊕ operator is σ_safe if Z alignment is preserved.
- ⊖ operator is potentially σ_unsafe but can be σ_safe if resulting chain
preserves the underlying morphic continuity Z.
```

σ Typing Hierarchy

σ_safe: $D(A_{n-1} \mid Z) \rightarrow D(B_{n-1} \mid Z) \mid Z$ (Preserves global coherence)
σ_unsafe: $D(A_{n-1} \mid Z) \rightarrow \{ \text{rupture}(f_1), \ldots, \text{rupture}(f_n) \mid \neg Z \}$ (Global coherence lost)

Note: σ_safe behaves as Quasi-Natural Transformation.
σ_unsafe induces rupture, and cannot be captured globally.

概念位相論 / Conceptual Topology

This theory, named 概念位相論 or Conceptual Topoloy, was proposed by **No Name Yet Exist**.

GitHub: https://github.com/No-Name-Yet-Exist/Conceptual-Topology
Note: https://note.com/xoreaxeax/n/n3711c1318d0b
Zenodo: https://zenodo.org/records/15455079

This is Version: 1.0 © 2025 No Name Yet Exist. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0). You may cite or reference this work with proper attribution. Commercial use, modification, or redistribution is prohibited without explicit permission.