

Continuous Gödel Encodings

Toward a Theory of Executable Symmetry

Adrian Diamond
AAD Systems
adrian@aadsystems.com

September 2026

Abstract

Classical Gödel encoding represents formal syntactic objects arithmetically so that operations on syntax can be reflected as operations on their encodings. This paper isolates the corresponding operational principle,

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f,$$

and extends it from discrete syntax to continuous transformation systems. A continuous Gödel encoding is developed as a faithful topological representation under which continuous transformations, group actions, composition laws, orbits, and invariants can be transported to an encoded state space. The resulting transformation structure is formulated as a continuous Gödel algebra.

Because transport by a homeomorphism is mathematically automatic, the paper distinguishes structural transport from effective execution. Using represented spaces, it gives a sufficient condition for effective transport: if the encoding, decoding, and source transformation possess computable realizers, then the encoded transformation

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is computable. Worked constructions for one-parameter actions, planar rotation, and finite-dimensional linear transformations illustrate the framework and the role of independently specified target representations. The construction generalizes an operational representation architecture associated with Gödel encoding; it does not assert a continuous analogue of Gödel's incompleteness theorems. The resulting framework poses a broader representation problem: determining which constrained representations preserve both the structural semantics and effective realizability of a transformation system.

1 Introduction

Gödel encoding provides a canonical example of a mathematical representation that is operational rather than merely descriptive [1]. Formal syntactic objects are assigned arithmetic representations, and syntactic transformations can themselves be represented as operations on the corresponding codes. At the structural level, this relation may be expressed by a commuting condition

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f,$$

where f acts on the original objects and $\widehat{f}$ acts on their representations.

This paper asks what remains of this architecture when the represented objects are no longer discrete syntactic structures but elements of a continuous state space. In particular, can a topological

transformation system be encoded so that its transformations, composition laws, symmetries, and invariants remain available on the encoded space? Under what additional conditions can those encoded transformations be executed effectively?

These questions lead to the notion of a *continuous Gödel encoding*. Let X be a topological space and let

$$\text{Enc} : X \rightarrow E$$

be an injective continuous encoding into a representation space E . When Enc is a homeomorphism onto its image

$$X_{\text{Enc}} := \text{Enc}(X),$$

the inverse map

$$\text{Dec} : X_{\text{Enc}} \rightarrow X$$

permits a continuous transformation

$$f : X \rightarrow X$$

to be transported to

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

The resulting transformation satisfies

$$\hat{f} \circ \text{Enc} = \text{Enc} \circ f.$$

At the purely topological level, this construction is a conjugation and is therefore structurally straightforward. This observation is important because it identifies the boundary between automatic transport and the stronger problem studied here. The existence of a continuous encoded transformation does not imply that the transformation is computable, nor does transport into an arbitrarily chosen homeomorphic representation constitute a substantive computational result.

The central distinction of the paper is therefore between *continuous transport* and *effective transport*. To make the latter precise, the framework uses represented spaces. When the encoding, decoding, and source transformation admit computable realizers relative to specified representations, the transported map

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is computable by closure under composition. This provides a precise sufficient condition for what we call *executable symmetry*.

The framework is developed first for individual transformations and then for transformation families and group actions. Encoding transport preserves identity and composition and, for invertible transformations, preserves inverses. Group actions transport to the encoded state space while preserving their action laws, orbit structure, and transported invariants. These operations form what we call a *continuous Gödel algebra*: the transformation algebra induced on the encoded state space by faithful representation.

Worked constructions illustrate the distinction between structural and effective transport. For planar rotation, an independently specified angular representation converts the geometric action

$$R_\theta$$

on the circle into

$$\hat{R}_\theta([\phi]) = [\phi + \theta],$$

so that rotation becomes addition modulo 2π . A finite-dimensional linear example similarly transports a matrix action A under a change of representation P to

$$\hat{A} = PAP^{-1}.$$

These examples also clarify the nontriviality condition: conjugation itself is automatic, while computational significance depends on the target representation and its effective constraints.

The terminology *continuous Gödel encoding* is intended to describe this extension of the operational representation principle motivating executable Gödel encoding. It is not a claim that topological or continuous systems satisfy analogues of Gödel’s incompleteness theorems. The commuting condition itself belongs to the broader mathematics of conjugacy and equivariance. The question specific to the present framework is whether a transformation system can be represented under specified constraints while preserving both its structural semantics and its effective realization.

The development proceeds as follows. Section 2 isolates the operational structure of executable Gödel encoding. Section 3 extends the encoding framework to topological spaces. Sections 4–7 develop transformation actions, equivariance, invariants, and the resulting continuous Gödel algebra. Section 8 introduces represented spaces and the effective transport condition. Section 9 gives worked constructions. Section 10 relates the framework to classical Gödel encoding, Section 11 identifies formalization and verification targets, and Section 12 places the construction relative to adjacent areas of mathematics and computation. Sections 13 and 14 discuss the limitations, open problems, and consequences of the framework.

The resulting hierarchy is

representation $\longrightarrow$ structural transport $\longrightarrow$ symmetry preservation $\longrightarrow$ effective realization.
--

This hierarchy leads to the broader problem motivating further work: given a transformation system together with independently specified computational constraints, determine which representations make its structure executable.

Notation

Throughout the paper, the following notation is used consistently.

Symbol	Meaning
X	source state or object space
E	ambient encoding or representation space
$X_{\text{Enc}} = \text{Enc}(X)$	encoded state space
Enc	encoding map
Dec	decoding map on X_{Enc}
f, g	transformations of the source space
$\hat{f}, \hat{g}$	corresponding encoded transformations
$\mathcal{F}$	family of source transformations
$\hat{\mathcal{F}}$	encoded transformation family
G	transformation group
α	group action on the source space
$\hat{\alpha}$	corresponding encoded group action
I	invariant on the source space
$\hat{I}$	transported invariant
δ_X	representation map for a represented space
$\mathfrak{G}_{\text{Enc}}(X, \mathcal{F})$	continuous Gödel transformation algebra

Unless otherwise specified, “faithful continuous encoding” means that $\text{Enc} : X \rightarrow X_{\text{Enc}}$ is a homeomorphism onto its image, with inverse $\text{Dec} : X_{\text{Enc}} \rightarrow X$.

The central transport relation is

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f,$$

and, for a faithful encoding,

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

2 Executable Gödel Encodings

2.1 Syntax and Arithmetic Encoding

Definition 2.1 (Discrete arithmetic coding and effective Gödel numbering). *Let Syn be a countable class of finite formal syntactic objects. An arithmetic coding, considered as a set-theoretic representation, is an injective map*

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N}.$$

Its image $\text{Enc}(\text{Syn}) \subseteq \mathbb{N}$ consists of valid codes. Injectivity alone supplies a set-theoretic partial inverse

$$\text{Dec} : \text{Enc}(\text{Syn}) \rightarrow \text{Syn}, \quad \text{Dec}(\text{Enc}(s)) = s,$$

but does not make either direction computable.

Suppose in addition that Syn is equipped with an effective syntax presentation. We call such a coding an effective Gödel numbering when encoding is computable and decoding is computable on valid codes, with the syntax and code operations needed for the intended application effectively realizable. In ordinary formal syntax these include effective formation and proof-checking relations; stronger arithmetical applications may require representability of the relevant syntactic relations in a specified arithmetic theory.

The distinction is essential: the commuting transport identities below hold for any injection, whereas computational claims about Gödel numbering require the additional effectiveness assumptions. This paper abstracts the operational architecture developed for executable syntactic coding in [4]; it does not identify an arbitrary injection with a classical effective Gödel numbering.

2.2 Encoding and Decoding

Definition 2.2 (Operational Encoding System). *Let X be a class of objects, let R be a representation space, and let*

$$\text{Enc} : X \rightarrow R$$

be injective. Its decoding map on the encoded image is

$$\text{Dec} := \text{Enc}^{-1} : \text{Enc}(X) \rightarrow X,$$

so that $\text{Dec}(\text{Enc}(x)) = x$ for every $x \in X$. Let $\mathcal{F}$ be a class of transformations

$$f : X \rightarrow X.$$

An operational encoding system for $(X, \mathcal{F})$ consists of Enc , Dec , and an assignment

$$f \mapsto \widehat{f},$$

where

$$\widehat{f} : \text{Enc}(X) \rightarrow \text{Enc}(X)$$

satisfies

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f}$$

for every $f \in \mathcal{F}$.

Equivalently, for every $x \in X$,

$$\widehat{f}(\text{Enc}(x)) = \text{Enc}(f(x)).$$

When decoding is available, the encoded transformation is necessarily given on $\text{Enc}(X)$ by

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

2.3 Encoded Transformations

Let $f : \text{Syn} \rightarrow \text{Syn}$ be a transformation of formal syntactic objects. Given an injective Gödel encoding

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N},$$

the corresponding encoded transformation is the map

$$\widehat{f} : \text{Enc}(\text{Syn}) \rightarrow \text{Enc}(\text{Syn})$$

defined by

$$\widehat{f} := \text{Enc} \circ f \circ \text{Dec}.$$

Consequently,

$$\widehat{f}(\text{Enc}(s)) = \text{Enc}(f(s))$$

for every $s \in \text{Syn}$.

Thus the encoded transformation does not merely associate a number with a syntactic object. It transports an operation on syntax into an operation on the representation of syntax.

The resulting square commutes:

$$\begin{array}{ccc} \text{Syn} & \xrightarrow{f} & \text{Syn} \\ \downarrow \text{Enc} & & \downarrow \text{Enc} \\ \text{Enc}(\text{Syn}) & \xrightarrow{\widehat{f}} & \text{Enc}(\text{Syn}). \end{array}$$

2.4 The Discrete Transformation Principle

The structural property that will be retained in the continuous setting is not the use of natural numbers itself, but the transport of transformations across an encoding.

More generally, suppose

$$\text{Enc} : X \rightarrow R$$

is injective and $f : X \rightarrow X$ is a transformation. A transformation

$$\widehat{f} : \text{Enc}(X) \rightarrow \text{Enc}(X)$$

represents f under Enc when

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

If $\text{Dec} = \text{Enc}^{-1}$ is defined on $\text{Enc}(X)$, then this condition determines $\hat{f}$ uniquely:

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

This commuting relation is the discrete transformation principle that will be generalized below. Classical Gödel numbering realizes it with arithmetic codes. The continuous theory replaces the discrete representation space by a topological representation space and asks which additional topological, algebraic, and effective structures are preserved under the transport.

3 From Discrete to Continuous Encoding

3.1 Why Arithmetic Coding Is Not Enough

Classical Gödel encoding is adapted to countable collections of finite syntactic objects. An injective map

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N}$$

is sufficient to distinguish such objects and to transport syntactic operations to their arithmetic representatives.

The present problem is different. Suppose that the source objects form a topological space X and that the transformations of interest are continuous maps

$$g : X \rightarrow X.$$

An arbitrary injection of the underlying set of X into another set may preserve identity of objects while destroying the topological structure that makes continuity meaningful. In that case the representation distinguishes objects but does not faithfully represent the continuous transformation system.

The relevant generalization therefore cannot consist merely of replacing one collection of codes by another. The encoding itself must interact with the topology. In particular, we seek a representation

$$\text{Enc} : X \rightarrow E$$

for which the source topology is retained on the encoded image and continuous transformations of X induce continuous transformations of $\text{Enc}(X)$.

Thus the transition from discrete to continuous encoding changes the representation problem from

distinguish and manipulate encoded objects

to

distinguish objects while preserving their continuous transformation structure.

3.2 Continuous Representation Spaces

Definition 3.1 (Continuous Encoding Space). *Let X be a topological space. A continuous encoding space for X is a topological space E together with an injective continuous map*

$$\text{Enc} : X \rightarrow E.$$

The encoded state space is the subspace

$$X_{\text{Enc}} := \text{Enc}(X) \subseteq E$$

equipped with the subspace topology inherited from E .

The encoding is called topologically faithful if

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

is a homeomorphism. In that case there exists a continuous decoding map

$$\text{Dec} := \text{Enc}^{-1} : X_{\text{Enc}} \rightarrow X,$$

and the representation preserves the topology of X exactly up to homeomorphism.

3.3 Continuous Encoding and Decoding

Definition 3.2 (Continuous Gödel Encoding). *Let X be a topological space of objects, let E be a topological encoding space, and let*

$$\text{Enc} : X \rightarrow E$$

be an injective continuous map.

Let $\mathcal{G}$ be a class of continuous transformations

$$g : X \rightarrow X.$$

A continuous Gödel encoding of $(X, \mathcal{G})$ consists of Enc together with an assignment

$$g \mapsto \widehat{g},$$

where each

$$\widehat{g} : \text{Enc}(X) \rightarrow \text{Enc}(X)$$

is continuous and satisfies

$$\boxed{\widehat{g} \circ \text{Enc} = \text{Enc} \circ g}$$

for every $g \in \mathcal{G}$.

If Enc is a homeomorphism onto its image, then the decoding map

$$\text{Dec} := \text{Enc}^{-1} : \text{Enc}(X) \rightarrow X$$

is continuous, and the encoded transformation is equivalently

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}.$$

3.4 Faithfulness and Recoverability

The preceding definitions separate two properties that should not be conflated. Injectivity guarantees that represented objects remain distinguishable. Topological faithfulness additionally guarantees that passage to the encoded image does not alter the topology of the source space.

Definition 3.3 (Faithful Continuous Encoding). *Let $(X, \mathcal{G})$ admit a continuous Gödel encoding*

$$\text{Enc} : X \rightarrow E.$$

The encoding is called faithful if

$$\text{Enc} : X \rightarrow \text{Enc}(X)$$

is a homeomorphism.

It is called transformation-faithful for $\mathcal{G}$ if the assignment

$$g \mapsto \widehat{g}$$

is injective: whenever $g, h \in \mathcal{G}$ satisfy

$$\widehat{g} = \widehat{h},$$

then

$$g = h.$$

Proposition 3.4 (Recoverability). *Let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous Gödel encoding. Then every encoded state $z \in \text{Enc}(X)$ determines a unique source state

$$x = \text{Dec}(z),$$

and every encoded transformation $\widehat{g}$ determines its source transformation by

$$g = \text{Dec} \circ \widehat{g} \circ \text{Enc}.$$

Proof. Since $\text{Enc} : X \rightarrow \text{Enc}(X)$ is a homeomorphism, its inverse

$$\text{Dec} : \text{Enc}(X) \rightarrow X$$

exists and is continuous. Hence every $z \in \text{Enc}(X)$ has the unique preimage $\text{Dec}(z)$.

For a represented transformation g , the defining commuting condition gives

$$\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.$$

Composing on the left with Dec yields

$$\text{Dec} \circ \widehat{g} \circ \text{Enc} = \text{Dec} \circ \text{Enc} \circ g = g.$$

□

Proposition 3.5 (Uniqueness of the Encoded Transformation). *Let $\text{Enc} : X \rightarrow E$ be injective, and let $g : X \rightarrow X$. There is at most one map*

$$\widehat{g} : \text{Enc}(X) \rightarrow \text{Enc}(X)$$

satisfying

$$\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.$$

If $\text{Dec} = \text{Enc}^{-1}$ is defined on $\text{Enc}(X)$, that map is

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}.$$

Proof. Let $z \in \text{Enc}(X)$. Then $z = \text{Enc}(x)$ for a unique $x \in X$. Any map satisfying the commuting condition must obey

$$\widehat{g}(z) = \widehat{g}(\text{Enc}(x)) = \text{Enc}(g(x)).$$

Thus its value is uniquely determined at every point of $\text{Enc}(X)$. Substituting $x = \text{Dec}(z)$ gives

$$\widehat{g}(z) = \text{Enc}(g(\text{Dec}(z))),$$

which is precisely

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}.$$

□

Corollary 3.6 (Faithfulness of Transport). *A faithful continuous encoding is transformation-faithful for any represented family of maps $g : X \rightarrow X$.*

Proof. Suppose

$$\widehat{g} = \widehat{h}.$$

By recoverability,

$$g = \text{Dec} \circ \widehat{g} \circ \text{Enc} = \text{Dec} \circ \widehat{h} \circ \text{Enc} = h.$$

□

4 Transformation Groups and Executable Symmetry

The preceding construction transports individual transformations across a faithful encoding. We now consider families of transformations carrying group structure. This provides the natural setting for symmetry: a symmetry is represented not merely by an isolated map, but by an action whose identity, composition, and inverse structure must survive encoding.

4.1 Group Actions

Definition 4.1 (Transformation Action). *Let G be a topological group with identity element e_G , and let X be a topological space. A continuous transformation action of G on X is a continuous map*

$$\alpha : G \times X \rightarrow X, \quad (g, x) \mapsto g \cdot x,$$

such that

$$e_G \cdot x = x$$

and

$$g \cdot (h \cdot x) = (gh) \cdot x$$

for every $g, h \in G$ and $x \in X$.

For each $g \in G$, define

$$\alpha_g : X \rightarrow X, \quad \alpha_g(x) = g \cdot x.$$

Then

$$\alpha_{e_G} = \text{id}_X$$

and

$$\alpha_g \circ \alpha_h = \alpha_{gh}.$$

Every α_g is a homeomorphism. Indeed,

$$\alpha_{g^{-1}} \circ \alpha_g = \alpha_{e_G} = \text{id}_X,$$

and similarly

$$\alpha_g \circ \alpha_{g^{-1}} = \text{id}_X.$$

Thus

$$\alpha_g^{-1} = \alpha_{g^{-1}}.$$

4.2 Lie Group Actions

When G is a Lie group and X is a smooth manifold, the same construction may be strengthened by requiring

$$\alpha : G \times X \rightarrow X$$

to be smooth. The resulting action is then a smooth Lie group action.

The topological theory developed below does not require differentiable structure. Lie group actions therefore form an important special case rather than an additional assumption of the general framework. When the encoding and decoding maps are diffeomorphisms onto the encoded image, the transported action also preserves the corresponding smooth structure.

4.3 Encoded Group Actions

Let

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and write

$$X_{\text{Enc}} := \text{Enc}(X).$$

For a continuous action

$$\alpha : G \times X \rightarrow X,$$

define the transported action

$$\hat{\alpha} : G \times X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

by

$$\boxed{\hat{\alpha}(g, z) := \text{Enc}(\alpha(g, \text{Dec}(z)))}.$$

Equivalently,

$$\hat{\alpha} = \text{Enc} \circ \alpha \circ (\text{id}_G \times \text{Dec}).$$

Definition 4.2 (Encoded Action). *An encoded action of G on X_{Enc} is an action*

$$\hat{\alpha} : G \times X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

satisfying

$$\boxed{\hat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(\alpha(g, x))}$$

for every $g \in G$ and $x \in X$.

Theorem 4.3 (Transport of Group Actions). *Let G be a topological group acting continuously on a topological space X , and let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding. Then

$$\hat{\alpha}(g, z) = \text{Enc}(\alpha(g, \text{Dec}(z)))$$

defines a continuous action of G on $X_{\text{Enc}} = \text{Enc}(X)$. Moreover,

$$\hat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(g \cdot x).$$

Proof. Since α , Enc , and Dec are continuous, the composition

$$\hat{\alpha} = \text{Enc} \circ \alpha \circ (\text{id}_G \times \text{Dec})$$

is continuous.

For the identity element,

$$\begin{aligned} \hat{\alpha}(e_G, z) &= \text{Enc}(e_G \cdot \text{Dec}(z)) \\ &= \text{Enc}(\text{Dec}(z)) \\ &= z. \end{aligned}$$

For $g, h \in G$,

$$\begin{aligned} \hat{\alpha}(g, \hat{\alpha}(h, z)) &= \text{Enc}(g \cdot \text{Dec}(\text{Enc}(h \cdot \text{Dec}(z)))) \\ &= \text{Enc}(g \cdot (h \cdot \text{Dec}(z))) \\ &= \text{Enc}((gh) \cdot \text{Dec}(z)) \\ &= \hat{\alpha}(gh, z). \end{aligned}$$

Hence $\hat{\alpha}$ satisfies the group-action laws.

Finally, for $z = \text{Enc}(x)$,

$$\begin{aligned} \hat{\alpha}(g, \text{Enc}(x)) &= \text{Enc}(g \cdot \text{Dec}(\text{Enc}(x))) \\ &= \text{Enc}(g \cdot x). \end{aligned}$$

□

Corollary 4.4 (Preservation of Symmetry Composition). *For each $g \in G$, define*

$$\hat{\alpha}_g(z) := \hat{\alpha}(g, z).$$

Then

$$\begin{aligned} \hat{\alpha}_{e_G} &= \text{id}_{X_{\text{Enc}}}, \\ \hat{\alpha}_g \circ \hat{\alpha}_h &= \hat{\alpha}_{gh}, \end{aligned}$$

and

$$\hat{\alpha}_{g^{-1}} = \hat{\alpha}_g^{-1}.$$

Proof. The first two identities are the group-action laws established above. For the third,

$$\hat{\alpha}_g \circ \hat{\alpha}_{g^{-1}} = \hat{\alpha}_{e_G} = \text{id}_{X_{\text{Enc}}},$$

and likewise in the reverse order.

□

4.4 Executable Symmetry

Transport alone establishes the existence of an encoded symmetry, but it does not establish that the symmetry is computationally executable. Indeed, if Enc is a homeomorphism onto its image, then

$$\hat{\alpha} = \text{Enc} \circ \alpha \circ (\text{id}_G \times \text{Dec})$$

exists automatically. Consequently, the existence of this transported action is not by itself the substantive executability condition.

Executability requires an additional effective structure on the spaces and maps involved. We therefore separate the topological notion established in this section from the computational notion that will be specified in Section 8.

Definition 4.5 (Executable Symmetry). *Let G act continuously on X , let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and let

$$\hat{\alpha} : G \times X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

be the transported encoded action.

The symmetry is called executable relative to the encoding if, in addition to satisfying

$$\hat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(g \cdot x),$$

the spaces G , X , and X_{Enc} are equipped with specified effective representations under which the required encoding, decoding, and action operations are computable.

The precise computational meaning of this condition depends on the chosen representation of continuous objects and is developed separately in Section 8.

Remark 4.6. The distinction between transported symmetry and executable symmetry is essential. A homeomorphic change of representation automatically transports a continuous group action. The additional mathematical content of executable symmetry begins only when the target representation is constrained independently and the transported operations are required to satisfy a specified notion of effective computation.

5 Equivariance and Semantic Preservation

The defining relation of a continuous Gödel encoding is an equivariance condition: transformation before encoding must agree with transformation after encoding. This section isolates that condition and records its consequences independently of the group-action setting.

5.1 Equivariant Encoding

Definition 5.1 (Equivariant Encoding). *Let X be a topological space, let*

$$\text{Enc} : X \rightarrow E$$

be a continuous encoding, and let

$$g : X \rightarrow X$$

and

$$\widehat{g} : \text{Enc}(X) \rightarrow \text{Enc}(X)$$

be transformations.

The encoding Enc is equivariant with respect to g and $\widehat{g}$ if

$$\boxed{\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.}$$

Equivalently,

$$\widehat{g}(\text{Enc}(x)) = \text{Enc}(g(x))$$

for every $x \in X$.

For a family $\mathcal{G}$ of transformations on X , the encoding is equivariant with respect to $\mathcal{G}$ if each $g \in \mathcal{G}$ admits a corresponding encoded transformation $\widehat{g}$ satisfying the same commuting relation.

In the special case of a group action

$$\alpha : G \times X \rightarrow X,$$

equivariance takes the form

$$\widehat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(\alpha(g, x)) = \text{Enc}(g \cdot x).$$

5.2 Commuting Transformation Structure

The equivariance condition is represented by the commuting diagram

$$\begin{array}{ccc} X & \xrightarrow{g} & X \\ \downarrow \text{Enc} & & \downarrow \text{Enc} \\ \text{Enc}(X) & \xrightarrow{\widehat{g}} & \text{Enc}(X). \end{array}$$

This diagram expresses the central representation principle of the construction. There are two paths from a source object x to an encoded transformed object. One may first transform x and then encode it,

$$x \xrightarrow{g} g(x) \xrightarrow{\text{Enc}} \text{Enc}(g(x)),$$

or first encode x and then apply the encoded transformation,

$$x \xrightarrow{\text{Enc}} \text{Enc}(x) \xrightarrow{\widehat{g}} \widehat{g}(\text{Enc}(x)).$$

Equivariance states that these paths agree.

When Enc is faithful, this agreement is exact rather than merely an approximation or correspondence: the encoded transformation is the conjugate

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}.$$

5.3 Transformation Preservation

Theorem 5.2 (Transformation Preservation). *Let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and let

$$g : X \rightarrow X$$

be continuous. Define

$$\widehat{g} := \text{Enc} \circ g \circ \text{Dec}$$

on $\text{Enc}(X)$. Then $\widehat{g}$ is continuous and satisfies

$$\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.$$

Moreover, g can be recovered from $\widehat{g}$ by

$$g = \text{Dec} \circ \widehat{g} \circ \text{Enc}.$$

Proof. Because Enc , g , and Dec are continuous, their composition

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}$$

is continuous.

Furthermore,

$$\begin{aligned} \widehat{g} \circ \text{Enc} &= \text{Enc} \circ g \circ \text{Dec} \circ \text{Enc} \\ &= \text{Enc} \circ g, \end{aligned}$$

since

$$\text{Dec} \circ \text{Enc} = \text{id}_X.$$

For recoverability,

$$\begin{aligned} \text{Dec} \circ \widehat{g} \circ \text{Enc} &= \text{Dec} \circ \text{Enc} \circ g \circ \text{Dec} \circ \text{Enc} \\ &= g. \end{aligned}$$

□

Corollary 5.3 (Preservation of Invertibility). *Under the hypotheses of the preceding theorem, if g is a homeomorphism, then $\widehat{g}$ is a homeomorphism and*

$$\widehat{g}^{-1} = \widehat{g^{-1}}.$$

Proof. Using the transported form,

$$\begin{aligned} \widehat{g} \circ \widehat{g^{-1}} &= \text{Enc} \circ g \circ \text{Dec} \circ \text{Enc} \circ g^{-1} \circ \text{Dec} \\ &= \text{Enc} \circ g \circ g^{-1} \circ \text{Dec} \\ &= \text{Enc} \circ \text{Dec} \\ &= \text{id}_{\text{Enc}(X)}. \end{aligned}$$

The reverse composition is identical.

□

5.4 Composition

Theorem 5.4 (Compositionality). *Let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and let

$$f, g : X \rightarrow X$$

be continuous transformations. If

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}$$

and

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec},$$

then

$$\boxed{\widehat{f \circ g} = \widehat{f} \circ \widehat{g}.}$$

Proof. By definition,

$$\begin{aligned} \widehat{f} \circ \widehat{g} &= \text{Enc} \circ f \circ \text{Dec} \circ \text{Enc} \circ g \circ \text{Dec} \\ &= \text{Enc} \circ f \circ (\text{Dec} \circ \text{Enc}) \circ g \circ \text{Dec} \\ &= \text{Enc} \circ f \circ g \circ \text{Dec} \\ &= \widehat{f \circ g}. \end{aligned}$$

□

Corollary 5.5 (Identity Preservation). *For the identity transformation*

$$\text{id}_X : X \rightarrow X,$$

the corresponding encoded transformation is

$$\widehat{\text{id}_X} = \text{id}_{\text{Enc}(X)}.$$

Proof. By transport,

$$\widehat{\text{id}_X} = \text{Enc} \circ \text{id}_X \circ \text{Dec} = \text{Enc} \circ \text{Dec} = \text{id}_{\text{Enc}(X)}.$$

□

Remark 5.6. The preceding results show that faithful continuous encoding preserves the transformation structure of the source system under conjugation. This fact alone should not be interpreted as a new computational result: it follows from the existence of the homeomorphism between X and its encoded image. The additional question addressed later is whether the encoded representation supports independently specified effective operations and therefore admits executable realization.

6 Invariants and Geometric Structure

Transporting transformations across an encoding preserves the transformation system up to conjugation. A stronger structural question is what quantities, equivalence relations, orbit structures, and geometric data associated with those transformations survive the change of representation.

The answer depends on the structure carried by the source and on the strength of the encoding. A homeomorphic encoding preserves topological structure, but it does not by itself preserve metric, smooth, symplectic, or other additional geometric structure. Such claims require corresponding hypotheses on the encoding.

6.1 Invariant Quantities

Let $\mathcal{G}$ be a family of transformations

$$g : X \rightarrow X.$$

Definition 6.1 (Encoding Invariant). *Let Y be a set and let*

$$I : X \rightarrow Y$$

be a quantity on the source space. The quantity I is invariant under $\mathcal{G}$ if

$$I(g(x)) = I(x)$$

for every $g \in \mathcal{G}$ and $x \in X$.

If

$$\text{Enc} : X \rightarrow E$$

is faithful with decoding map

$$\text{Dec} : \text{Enc}(X) \rightarrow X,$$

the corresponding encoded invariant is

$$\widehat{I} : \text{Enc}(X) \rightarrow Y$$

defined by

$$\boxed{\widehat{I} := I \circ \text{Dec}.}$$

Thus

$$\widehat{I}(\text{Enc}(x)) = I(x),$$

so the value of the invariant can be recovered directly from the encoded state.

Proposition 6.2 (Invariant Transport). *Let $I : X \rightarrow Y$ be invariant under $g : X \rightarrow X$, and let*

$$\widehat{g} = \text{Enc} \circ g \circ \text{Dec}$$

be its transported transformation. Then

$$\widehat{I} = I \circ \text{Dec}$$

is invariant under $\widehat{g}$:

$$\boxed{\widehat{I} \circ \widehat{g} = \widehat{I}.}$$

Proof. For $z \in \text{Enc}(X)$,

$$\begin{aligned} \widehat{I}(\widehat{g}(z)) &= I(\text{Dec}(\text{Enc}(g(\text{Dec}(z))))) \\ &= I(g(\text{Dec}(z))) \\ &= I(\text{Dec}(z)) \\ &= \widehat{I}(z). \end{aligned}$$

Hence

$$\widehat{I} \circ \widehat{g} = \widehat{I}.$$

□

6.2 Orbits and Equivalence Classes

Suppose now that a group G acts on X . For $x \in X$, its orbit is

$$G \cdot x := \{g \cdot x : g \in G\}.$$

The action induces an equivalence relation

$$x \sim_G y$$

whenever there exists $g \in G$ such that

$$y = g \cdot x.$$

The transported action on

$$X_{\text{Enc}} := \text{Enc}(X)$$

induces the corresponding encoded orbit

$$G \cdot \text{Enc}(x) := \{\hat{\alpha}(g, \text{Enc}(x)) : g \in G\}.$$

Proposition 6.3 (Orbit Preservation). *Let G act on X , and let Enc be a faithful encoding with transported action $\hat{\alpha}$. Then*

$$\boxed{G \cdot \text{Enc}(x) = \text{Enc}(G \cdot x).}$$

Consequently,

$$x \sim_G y \iff \text{Enc}(x) \sim_G \text{Enc}(y).$$

Proof. By equivariance,

$$\hat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(g \cdot x).$$

Therefore

$$\begin{aligned} G \cdot \text{Enc}(x) &= \{\hat{\alpha}(g, \text{Enc}(x)) : g \in G\} \\ &= \{\text{Enc}(g \cdot x) : g \in G\} \\ &= \text{Enc}(G \cdot x). \end{aligned}$$

If $x \sim_G y$, then $y = g \cdot x$ for some $g \in G$, and hence

$$\text{Enc}(y) = \hat{\alpha}(g, \text{Enc}(x)),$$

so

$$\text{Enc}(x) \sim_G \text{Enc}(y).$$

Conversely, if

$$\text{Enc}(y) = \hat{\alpha}(g, \text{Enc}(x)),$$

then equivariance gives

$$\text{Enc}(y) = \text{Enc}(g \cdot x).$$

Injectivity of Enc implies

$$y = g \cdot x,$$

and therefore

$$x \sim_G y.$$

□

6.3 Invariant Geometric Structure

A faithful continuous encoding preserves the topology of X because

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

is a homeomorphism. Additional geometric structure, however, is preserved only when the encoding is required to respect that structure.

This distinction can be expressed by considering a structure $\mathcal{S}$ carried by X . Examples include a metric, smooth structure, measure, differential form, or other geometric datum. To claim preservation of $\mathcal{S}$, one must specify a corresponding structure

$$\hat{\mathcal{S}}$$

on X_{Enc} and require Enc to be an isomorphism in the relevant category.

For example, if (X, d) is a metric space, define

$$\hat{d}(z_1, z_2) := d(\text{Dec}(z_1), \text{Dec}(z_2))$$

for $z_1, z_2 \in X_{\text{Enc}}$. Then

$$\hat{d}(\text{Enc}(x), \text{Enc}(y)) = d(x, y),$$

so

$$\text{Enc} : (X, d) \rightarrow (X_{\text{Enc}}, \hat{d})$$

is an isometry.

Similarly, when X and X_{Enc} carry smooth structures and Enc is a diffeomorphism onto its image, smooth geometric objects may be transported through Enc and Dec . Such preservation is therefore conditional on the structure imposed on the representation, not a consequence of continuity alone.

6.4 Preservation Across Representation

Theorem 6.4 (Invariant Preservation). *Let G act on X , let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and let

$$\hat{\alpha}$$

be the transported action on X_{Enc} .

Suppose

$$I : X \rightarrow Y$$

is invariant under the action of G , so that

$$I(g \cdot x) = I(x)$$

for all $g \in G$ and $x \in X$. Define

$$\hat{I} := I \circ \text{Dec}.$$

Then $\hat{I}$ is invariant under the encoded action:

$$\boxed{\hat{I}(\hat{\alpha}(g, z)) = \hat{I}(z)}$$

for every $g \in G$ and $z \in X_{\text{Enc}}$.

Proof. For $z \in X_{\text{Enc}}$,

$$\begin{aligned}\widehat{I}(\widehat{\alpha}(g, z)) &= I(\text{Dec}(\text{Enc}(g \cdot \text{Dec}(z)))) \\ &= I(g \cdot \text{Dec}(z)) \\ &= I(\text{Dec}(z)) \\ &= \widehat{I}(z).\end{aligned}$$

□

Corollary 6.5 (Invariant Separation of Encoded Orbits). *If an invariant*

$$I : X \rightarrow Y$$

takes different values at $x, y \in X$, then $\text{Enc}(x)$ and $\text{Enc}(y)$ cannot lie in the same encoded G -orbit.

Proof. If $\text{Enc}(x)$ and $\text{Enc}(y)$ belonged to the same encoded orbit, orbit preservation would imply that x and y belonged to the same source orbit. Since I is constant on every orbit, this would imply

$$I(x) = I(y),$$

contrary to the hypothesis. □

Remark 6.6. The results of this section separate two levels of preservation. Topological structure and transformation structure are preserved by a faithful continuous encoding. Additional geometric structure is preserved only under corresponding stronger hypotheses on the encoding. This distinction prevents the representation framework from attributing geometric content to an encoding that has not actually been shown to preserve it.

7 Continuous Gödel Algebra

The preceding sections considered encoded transformations individually and established preservation of composition, identity, inverses, group actions, and invariants. These operations can now be collected into a single algebraic structure.

Let

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and write

$$X_{\text{Enc}} := \text{Enc}(X).$$

Let $\mathcal{F}$ be a family of continuous transformations

$$f : X \rightarrow X.$$

7.1 Encoded Transformations

Definition 7.1 (Continuous Encoded Transformation). *For each $f \in \mathcal{F}$, the continuous encoded transformation associated with f is*

$$\widehat{f} : X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

defined by

$$\boxed{\widehat{f} := \text{Enc} \circ f \circ \text{Dec}.}$$

The family of encoded transformations associated with $\mathcal{F}$ is

$$\widehat{\mathcal{F}} := \{\widehat{f} : f \in \mathcal{F}\}.$$

The transport map

$$T_{\text{Enc}} : \mathcal{F} \rightarrow \widehat{\mathcal{F}}, \quad T_{\text{Enc}}(f) = \widehat{f},$$

will be called the *encoding transport*.

By construction,

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f$$

for every $f \in \mathcal{F}$.

Moreover, faithfulness of Enc implies that T_{Enc} is injective. Indeed,

$$\widehat{f} = \widehat{g} \implies f = \text{Dec} \circ \widehat{f} \circ \text{Enc} = \text{Dec} \circ \widehat{g} \circ \text{Enc} = g.$$

7.2 Composition Law

Suppose $\mathcal{F}$ is closed under composition. For $f, g \in \mathcal{F}$,

$$\begin{aligned} \widehat{f \circ g} &= \text{Enc} \circ f \circ \text{Dec} \circ \text{Enc} \circ g \circ \text{Dec} \\ &= \text{Enc} \circ f \circ g \circ \text{Dec} \\ &= \widehat{f \circ g}. \end{aligned}$$

Hence

$$\boxed{T_{\text{Enc}}(f \circ g) = T_{\text{Enc}}(f) \circ T_{\text{Enc}}(g).}$$

Thus encoding transport preserves the composition law exactly.

7.3 Identity

If

$$\text{id}_X \in \mathcal{F},$$

then its encoded transformation is

$$\begin{aligned} \widehat{\text{id}_X} &= \text{Enc} \circ \text{id}_X \circ \text{Dec} \\ &= \text{Enc} \circ \text{Dec} \\ &= \text{id}_{X_{\text{Enc}}}. \end{aligned}$$

Therefore

$$\boxed{T_{\text{Enc}}(\text{id}_X) = \text{id}_{X_{\text{Enc}}}.$$

The identity element of the source transformation system is therefore transported to the identity element of the encoded transformation system.

7.4 Inverse Transformations

Suppose $f \in \mathcal{F}$ is invertible and

$$f^{-1} \in \mathcal{F}.$$

Then

$$\begin{aligned} \widehat{f \circ f^{-1}} &= \widehat{f \circ f^{-1}} \\ &= \widehat{\text{id}_X} \\ &= \text{id}_{X_{\text{Enc}}}, \end{aligned}$$

and similarly

$$\widehat{f^{-1}} \circ \widehat{f} = \text{id}_{X_{\text{Enc}}}.$$

Consequently,

$$\boxed{\widehat{f^{-1}} = \widehat{f}^{-1}}.$$

Thus invertibility is preserved by encoding transport whenever the source family contains the relevant inverse transformations.

7.5 Algebraic Closure

Theorem 7.2 (Closure of Encoded Transformations). *Let*

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding, and let $\mathcal{F}$ be a family of continuous transformations of X that contains id_X and is closed under composition. Then

$$\widehat{\mathcal{F}} = \{\text{Enc} \circ f \circ \text{Dec} : f \in \mathcal{F}\}$$

contains $\text{id}_{X_{\text{Enc}}}$ and is closed under composition.

Moreover,

$$T_{\text{Enc}} : \mathcal{F} \rightarrow \widehat{\mathcal{F}}$$

is a bijection satisfying

$$T_{\text{Enc}}(\text{id}_X) = \text{id}_{X_{\text{Enc}}}$$

and

$$T_{\text{Enc}}(f \circ g) = T_{\text{Enc}}(f) \circ T_{\text{Enc}}(g).$$

Hence, whenever $\mathcal{F}$ is a monoid under composition, $\widehat{\mathcal{F}}$ is a monoid and

$$T_{\text{Enc}} : \mathcal{F} \cong \widehat{\mathcal{F}}$$

is a monoid isomorphism.

Proof. Since $\text{id}_X \in \mathcal{F}$,

$$T_{\text{Enc}}(\text{id}_X) = \text{id}_{X_{\text{Enc}}},$$

so the encoded family contains the identity.

Let

$$\widehat{f}, \widehat{g} \in \widehat{\mathcal{F}}.$$

Then $f, g \in \mathcal{F}$. Since $\mathcal{F}$ is closed under composition,

$$f \circ g \in \mathcal{F}.$$

By compositionality,

$$\widehat{f} \circ \widehat{g} = \widehat{f \circ g} \in \widehat{\mathcal{F}}.$$

Thus $\widehat{\mathcal{F}}$ is closed under composition.

The map T_{Enc} is surjective by the definition of $\widehat{\mathcal{F}}$, and it is injective by recoverability:

$$f = \text{Dec} \circ \widehat{f} \circ \text{Enc}.$$

Therefore T_{Enc} is a bijection. Preservation of identity and composition was established above, proving the monoid-isomorphism claim. $\square$

Corollary 7.3 (Group Transport). *If $\mathcal{F}$ is a group of homeomorphisms of X under composition, then $\widehat{\mathcal{F}}$ is a group of homeomorphisms of X_{Enc} and*

$$T_{\text{Enc}} : \mathcal{F} \cong \widehat{\mathcal{F}}$$

is a group isomorphism.

Proof. The preceding theorem gives identity, composition, closure, and the bijection T_{Enc} . Since every $f \in \mathcal{F}$ has $f^{-1} \in \mathcal{F}$,

$$T_{\text{Enc}}(f^{-1}) = T_{\text{Enc}}(f)^{-1}.$$

Hence inverses are also preserved. $\square$

Definition 7.4 (Continuous Gödel Algebra). *Let $\mathcal{F}$ be a transformation family on X equipped with composition and identity, and let $\text{Enc} : X \rightarrow E$ be a faithful continuous encoding.*

The continuous Gödel algebra associated with $(X, \mathcal{F}, \text{Enc})$ is the transported transformation system

$$\mathfrak{G}_{\text{Enc}}(X, \mathcal{F}) := (X_{\text{Enc}}, \widehat{\mathcal{F}}, \circ, \text{id}_{X_{\text{Enc}}}),$$

together with the encoding transport

$$T_{\text{Enc}} : \mathcal{F} \rightarrow \widehat{\mathcal{F}}.$$

When $\mathcal{F}$ is a monoid, this transport is a monoid isomorphism. When $\mathcal{F}$ is a group of homeomorphisms, it is a group isomorphism.

Remark 7.5. The term “algebra” here refers specifically to the algebraic structure of the transported transformation family. No ring, field, or vector-space structure is implied unless such additional operations are separately introduced and shown to be preserved.

Furthermore, the algebraic isomorphism above follows from faithful transport by conjugation. Its significance for executable symmetry therefore depends on the additional effective constraints imposed on the encoded representation, which are addressed in the next section.

8 Executability

The preceding sections establish that a faithful continuous encoding transports transformations, composition, inverses, group actions, and invariants from a source space to its encoded image. These results are structural. They do not by themselves imply that the encoded transformations can be executed by an algorithm.

Indeed, if

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

is a homeomorphism, then every continuous transformation

$$f : X \rightarrow X$$

has the transported form

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

This establishes existence and continuity, but not computability.

Executability therefore requires additional data: continuous objects must possess effective representations, and the relevant maps must admit effective procedures acting on those representations.

8.1 What “Executable” Means in the Continuous Setting

We use the standard represented-space distinction between an abstract object and a name from which that object can be computationally accessed [2].

Definition 8.1 (Represented Space). *A represented space is a pair*

$$(X, \delta_X),$$

where

$$\delta_X : \subseteq \mathbb{N}^{\mathbb{N}} \rightarrow X$$

is a partial surjection.

An element

$$p \in \text{dom}(\delta_X)$$

satisfying

$$\delta_X(p) = x$$

is called a name of x .

The use of $\mathbb{N}^{\mathbb{N}}$ is not intended to identify a continuous object with a single finite code. Rather, a name may provide successively refined finite information about the represented object.

Definition 8.2 (Computable Map Between Represented Spaces). *Let (X, δ_X) and (Y, δ_Y) be represented spaces. A map*

$$f : X \rightarrow Y$$

is computable relative to the representations δ_X and δ_Y if there exists a computable partial map

$$F : \subseteq \mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\mathbb{N}}$$

such that whenever

$$\delta_X(p) = x,$$

we have

$$\delta_Y(F(p)) = f(x).$$

Such an F is called a realizer of f . In particular, the realizer must be defined on every valid input name $p \in \text{dom}(\delta_X)$, and $F(p)$ must lie in $\text{dom}(\delta_Y)$. The map F may be undefined on names that are not valid inputs.

A represented space carries the final topology induced by its representation. For the topological and computational layers of this paper to describe the *same* spaces, we require that the chosen representations induce the topologies used in the preceding sections. Standard admissible representations provide a useful setting for this compatibility [6]; an arbitrary representation need not induce the originally specified topology. All computability claims below are relative to the explicitly selected representations.

Thus executability is not a property of an abstract continuous map alone. It is a property of the map together with specified representations of its domain and codomain.

8.2 Finite Representations of Continuous Objects

Although a name may be infinite, an executable procedure can inspect only finite portions of that name during any finite computation. Finite computation therefore proceeds through finite information about the represented object.

For metric spaces this idea may be made concrete through approximations. Suppose (X, d_X) has a designated countable family of representable approximants

$$(a_n)_{n \in \mathbb{N}}.$$

A name for $x \in X$ may encode a sequence of approximants converging effectively to x , for example with a prescribed error bound

$$d_X(a_{p(n)}, x) \leq 2^{-n}.$$

The particular representation is part of the computational structure and must be specified. Different representations of the same underlying topological space need not induce the same notion of effective computation.

Accordingly, a continuous Gödel encoding becomes computationally meaningful only after representations have been chosen for the source space, the encoded space, and any transformation parameters.

8.3 Approximation and Error

Let (X, d_X) and (Y, d_Y) be represented metric spaces, and let

$$f : X \rightarrow Y$$

be computable with respect to their chosen representations. Suppose additionally that the chosen representation of Y supports effective metric approximation by representable approximants.

An executable approximation procedure for f may then receive a representation of x together with a requested precision $n \in \mathbb{N}$ and produce a representable output y_n satisfying

$$d_Y(y_n, f(x)) \leq 2^{-n}.$$

The numerical bound 2^{-n} is a convenient precision convention; other effective error schedules may be used.

For an encoded transformation

$$\hat{f} : X_{\text{Enc}} \rightarrow X_{\text{Enc}},$$

the analogous requirement is that, from a name of an encoded state z and a requested precision, one can effectively obtain an approximation to

$$\hat{f}(z).$$

This separates three statements:

- Existence: $\hat{f}(z)$ is mathematically defined,
- Continuity: $\hat{f}$ preserves the relevant topological limits,
- Executability: $\hat{f}(z)$ can be computed from represented input.

Only the third statement supplies the additional effective content required by executable symmetry.

8.4 Executable Transformation Procedures

Let

$$(X, \delta_X)$$

and

$$(X_{\text{Enc}}, \delta_E)$$

be represented spaces. Suppose the encoding and decoding maps

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

and

$$\text{Dec} : X_{\text{Enc}} \rightarrow X$$

are computable relative to these representations.

Let

$$f : X \rightarrow X$$

also be computable. Then the transported transformation

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is computable, because computable realizers are closed under composition.

This yields the following basic result.

Theorem 8.3 (Effective Transport). *Let (X, δ_X) and $(X_{\text{Enc}}, \delta_E)$ be represented spaces. Suppose*

$$\text{Enc} : X \rightarrow X_{\text{Enc}}, \quad \text{Dec} : X_{\text{Enc}} \rightarrow X$$

are mutually inverse computable maps, and let

$$f : X \rightarrow X$$

be computable.

Then

$$\hat{f} := \text{Enc} \circ f \circ \text{Dec}$$

is computable and satisfies

$$\hat{f} \circ \text{Enc} = \text{Enc} \circ f.$$

Proof. Let

$$F_{\text{Enc}}, \quad F_f, \quad F_{\text{Dec}}$$

be computable realizers of Enc , f , and Dec , respectively. Their composition

$$F_{\text{Enc}} \circ F_f \circ F_{\text{Dec}}$$

is computable and realizes

$$\text{Enc} \circ f \circ \text{Dec} = \hat{f}.$$

The commuting identity follows from

$$\begin{aligned} \hat{f} \circ \text{Enc} &= \text{Enc} \circ f \circ \text{Dec} \circ \text{Enc} \\ &= \text{Enc} \circ f. \end{aligned}$$

□

The same principle applies to a parameterized group action. If G , X , and X_{Enc} carry effective representations and

$$\alpha : G \times X \rightarrow X, \quad \text{Enc} : X \rightarrow X_{\text{Enc}}, \quad \text{Dec} : X_{\text{Enc}} \rightarrow X$$

are computable, then

$$\hat{\alpha} = \text{Enc} \circ \alpha \circ (\text{id}_G \times \text{Dec})$$

is a computable encoded action.

Corollary 8.4 (Effective Transport of Symmetry). *Under the preceding hypotheses, if the source action*

$$\alpha : G \times X \rightarrow X$$

is computable relative to the specified representations, then the transported action

$$\hat{\alpha} : G \times X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

is computable and satisfies

$$\hat{\alpha}(g, \text{Enc}(x)) = \text{Enc}(g \cdot x).$$

8.5 Correctness Conditions

Executability and correctness are distinct. A procedure may compute some encoded transformation without computing the transformation specified by the source semantics. Correctness therefore requires the executable procedure to realize the transported map.

Definition 8.5 (Executable Correctness). *Let*

$$f : X \rightarrow X$$

be a source transformation and let

$$\hat{f} : X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

be an encoded transformation.

An executable procedure for $\hat{f}$ is correct relative to (f, Enc) if it computes $\hat{f}$ with respect to the specified representations and

$$\boxed{\hat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

Equivalently, for every $x \in X$,

$$\hat{f}(\text{Enc}(x)) = \text{Enc}(f(x)).$$

Theorem 8.6 (Executable Correctness by Effective Conjugation). *Suppose*

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

and

$$\text{Dec} : X_{\text{Enc}} \rightarrow X$$

are mutually inverse computable maps between represented spaces, and let

$$f : X \rightarrow X$$

be computable. Then the procedure realizing

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is executable and correct relative to (f, Enc) .

Proof. Executability follows from the Effective Transport theorem. Correctness follows from

$$\begin{aligned}\widehat{f} \circ \text{Enc} &= \text{Enc} \circ f \circ \text{Dec} \circ \text{Enc} \\ &= \text{Enc} \circ f.\end{aligned}$$

□

8.6 A Boundary Example: Topological Transport Need Not Be Effective

Proposition 8.7 (Noncomputable conjugation under a fixed representation). *Equip the source and target copies of $\mathbb{R}$ with their standard Cauchy representations. There exist a homeomorphism $\text{Enc} : \mathbb{R} \rightarrow \mathbb{R}$ and a computable continuous map $f : \mathbb{R} \rightarrow \mathbb{R}$ whose transported map $\widehat{f} = \text{Enc} \circ f \circ \text{Enc}^{-1}$ is not computable in the fixed target representation.*

Proof. Choose a real number a that is not computable in the standard Cauchy representation. Put

$$\text{Enc}(x) = x + a, \quad \text{Dec}(z) = z - a, \quad f(x) = 2x.$$

The map Enc is a homeomorphism and f is computable. However,

$$\widehat{f}(z) = \text{Enc}(2(z - a)) = 2z - a.$$

If $\widehat{f}$ were computable relative to the standard representation of its target, its value at the computable input 0 would be a computable real. Since $\widehat{f}(0) = -a$, this would make a computable, contradicting its choice. □

This counterexample uses the *same independently fixed* standard representation on both copies of $\mathbb{R}$. The failure does not come from noncontinuity: Enc and $\widehat{f}$ are continuous. It comes from an encoding whose topological inverse exists but whose numerical content is not effectively available. It is an elementary boundary illustration, not a new noncomputability phenomenon.

Remark 8.8. The effective-conjugation theorem identifies a sufficient condition for executable transport, but it also clarifies the remaining nontriviality requirement. If the effective representation on X_{Enc} is introduced solely by transporting every computational structure from X , then executability again follows essentially by construction.

A substantive executable encoding problem therefore specifies independent constraints on the target representation—for example, a fixed data representation, admissible primitive operations, a required approximation discipline, or a computational resource model—and asks whether the source transformations can still be realized within those constraints.

9 Worked Constructions

We now illustrate the preceding framework with explicit continuous transformation systems. The first construction isolates the general one-parameter case. The second gives a concrete effectively realizable representation of planar rotation. The third extends the same idea to finite-dimensional linear actions. Finally, we identify the discrete case as a special regime of the general representation principle.

9.1 A One-Parameter Transformation Group

Let X be a topological space and suppose

$$\varphi : \mathbb{R} \times X \rightarrow X$$

is a continuous action of the additive group $(\mathbb{R}, +)$. Thus

$$\varphi(0, x) = x$$

and

$$\varphi(s, \varphi(t, x)) = \varphi(s + t, x)$$

for every $s, t \in \mathbb{R}$ and $x \in X$.

Let

$$\text{Enc} : X \rightarrow E$$

be a faithful continuous encoding and write

$$X_{\text{Enc}} := \text{Enc}(X).$$

Define

$$\widehat{\varphi} : \mathbb{R} \times X_{\text{Enc}} \rightarrow X_{\text{Enc}}$$

by

$$\widehat{\varphi}(t, z) := \text{Enc}(\varphi(t, \text{Dec}(z))).$$

Then

$$\widehat{\varphi}(0, z) = z$$

and

$$\widehat{\varphi}(s, \widehat{\varphi}(t, z)) = \widehat{\varphi}(s + t, z).$$

Hence every continuous one-parameter transformation group is transported to a continuous one-parameter transformation group on the encoded state space.

If $\mathbb{R}$, X , and X_{Enc} are equipped with effective representations for which φ , Enc , and Dec are computable, then the Effective Transport theorem implies that $\widehat{\varphi}$ is computable as well.

9.2 Rotation as Encoded Symmetry

Consider the unit circle

$$S^1 = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}.$$

The rotation group

$$SO(2)$$

acts continuously on S^1 by matrix multiplication.

For $\theta \in \mathbb{R}$, write

$$R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Then

$$R_\theta \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \end{pmatrix}.$$

Rather than constructing the encoded space by an arbitrary homeomorphic copy of S^1 , fix the target representation independently as the angular representation

$$E_{\text{ang}} := \mathbb{R}/(2\pi\mathbb{Z}).$$

Define

$$\text{Enc} : S^1 \rightarrow E_{\text{ang}}$$

by assigning to each point its angle modulo 2π :

$$\text{Enc}(\cos \phi, \sin \phi) = [\phi].$$

The inverse map is

$$\text{Dec}([\phi]) = (\cos \phi, \sin \phi).$$

Under this encoding, rotation by θ becomes

$$\widehat{R}_\theta : E_{\text{ang}} \rightarrow E_{\text{ang}}$$

with

$$\boxed{\widehat{R}_\theta([\phi]) = [\phi + \theta].}$$

Indeed,

$$\begin{aligned} \widehat{R}_\theta(\text{Enc}(\cos \phi, \sin \phi)) &= [\phi + \theta] \\ &= \text{Enc} \left(R_\theta \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} \right). \end{aligned}$$

Thus

$$\widehat{R}_\theta \circ \text{Enc} = \text{Enc} \circ R_\theta.$$

Composition is represented by addition:

$$\begin{aligned} \widehat{R}_\theta \circ \widehat{R}_\psi([\phi]) &= [\phi + \psi + \theta] \\ &= \widehat{R}_{\theta+\psi}([\phi]). \end{aligned}$$

The inverse is

$$\widehat{R}_\theta^{-1} = \widehat{R}_{-\theta}.$$

This example also makes the effective question explicit. Equip $\mathbb{R}$ with a standard effective representation $\delta_{\mathbb{R}}$, such as one based on rapidly converging rational approximations, and equip

$$E_{\text{ang}} = \mathbb{R}/(2\pi\mathbb{Z})$$

with the induced quotient representation

$$\delta_{\text{ang}}(p) := [\delta_{\mathbb{R}}(p)].$$

Under this representation, the encoded rotation

$$[\phi] \longmapsto [\phi + \theta]$$

is computable whenever θ is supplied by a computable name, because a name of $\phi + \theta$ can be obtained by effective real addition and then interpreted through δ_{ang} .

To verify the stronger *executable symmetry* requirement, equip $S^1 \subseteq \mathbb{R}^2$ with the subspace Cauchy representation: a name of a point supplies computable names of its Cartesian coordinates (x, y) . With this representation, both maps **Enc** and **Dec** have computable realizers. For **Dec**, from a real name of ϕ compute the real names of $\cos \phi$ and $\sin \phi$. The result is independent of the chosen representative of $[\phi]$.

For **Enc**, a real-angle representative can be computed *locally* from the Cartesian coordinates. Run in parallel the semidecision procedures for

$$x > \frac{1}{2}, \quad x < -\frac{1}{2}, \quad y > \frac{1}{2}, \quad y < -\frac{1}{2}.$$

At least one eventually succeeds, since $x^2 + y^2 = 1$. For the first successful test, select respectively the real angle

$$\begin{aligned} \arctan(y/x), & \quad x > \frac{1}{2}, \\ \pi + \arctan(y/x), & \quad x < -\frac{1}{2}, \\ \frac{\pi}{2} - \arctan(x/y), & \quad y > \frac{1}{2}, \\ -\frac{\pi}{2} - \arctan(x/y), & \quad y < -\frac{1}{2}. \end{aligned}$$

Each selected expression is computable on its tested open region and represents the correct point modulo 2π . Taking its quotient name therefore realizes **Enc**. This does *not* assert that a single continuous global real-valued argument function exists on S^1 ; only a computable *quotient-valued* encoder is needed.

Consequently, under these specified representations, **Enc**, **Dec**, and the encoded rotation are all computable. The formula $[\phi] \mapsto [\phi + \theta]$ is uniformly computable when a name of θ is provided as an input. For a fixed-parameter implementation, the parameter θ must itself be a computable real. The encoded operation is therefore not merely known to exist through abstract conjugation: it is effectively realized by addition in the independently fixed angular representation.

9.3 A Higher-Dimensional Example

Let

$$X = \mathbb{R}^n$$

and let a matrix group

$$G \subseteq GL(n, \mathbb{R})$$

act by

$$A \cdot x = Ax.$$

Fix an invertible matrix

$$P \in GL(n, \mathbb{R})$$

and define the encoding

$$\text{Enc} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \text{Enc}(x) = Px.$$

Then

$$\text{Dec}(z) = P^{-1}z.$$

For $A \in G$, the encoded transformation is

$$\begin{aligned} \widehat{A}(z) &= \text{Enc}(A \text{Dec}(z)) \\ &= PAP^{-1}z. \end{aligned}$$

Thus

$$\boxed{\widehat{A} = PAP^{-1}.}$$

For $A, B \in G$,

$$\begin{aligned}\widehat{A}\widehat{B} &= PAP^{-1}PBP^{-1} \\ &= PABP^{-1} \\ &= \widehat{AB}.\end{aligned}$$

Hence the encoded transformation family is the conjugate matrix group

$$\widehat{G} = PGP^{-1}.$$

If P , P^{-1} , and the matrices in the chosen action have computable real entries under a specified representation, then the corresponding finite-dimensional matrix operations are computable. This provides a direct finite-dimensional instance of effective transport.

This example also illustrates the limitation emphasized earlier: conjugation by P alone is mathematically automatic. Additional content arises only when the target coordinates or permitted operations are fixed independently or when the change of representation exposes structure useful for computation, verification, or invariant analysis.

9.4 The Discrete Case as a Special Regime

The continuous framework contains the discrete case as a special regime.

Let X be a set equipped with the discrete topology. Every map

$$f : X \rightarrow X$$

is continuous. If

$$\text{Enc} : X \rightarrow E$$

is injective and $\text{Enc}(X)$ is also given the discrete topology, then

$$\text{Enc} : X \rightarrow \text{Enc}(X)$$

is a homeomorphism.

The transported transformation is therefore

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec},$$

with

$$\widehat{f}(\text{Enc}(x)) = \text{Enc}(f(x)).$$

If $X = \text{Syn}$ is a countable formal syntax and

$$E = \mathbb{N},$$

this reduces to the familiar Gödel-style representation

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N}.$$

The continuous theory therefore preserves the same commuting principle:

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

What changes is the structure demanded of the representation. In the discrete setting, arithmetic coding and effective manipulation of finite syntax are central. In the continuous setting, topology and effective representations of non-discrete objects must additionally be specified.

Remark 9.1. The phrase “discrete limit” is used here structurally rather than as an analytic limiting process. No convergence of a continuous family to a discrete system is asserted. The claim is that discrete spaces form a special case of the topological framework in which continuity places no additional restriction on transformations.

10 Relation to Classical Gödel Encoding

The term *continuous Gödel encoding* is intended to identify a structural generalization of the representation principle underlying classical Gödel numbering. The generalization concerns the encoding of objects together with operations on those objects. It does not assert a continuous analogue of Gödel’s incompleteness theorems.

This distinction is essential. Classical Gödel numbering operates on formal syntax and represents syntactic objects arithmetically. The present framework extracts the transport principle from that construction and studies its extension to topological transformation systems.

10.1 What Is Preserved

Let Syn be a collection of formal syntactic objects and let

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N}$$

be a classical Gödel encoding. For a syntactic transformation

$$f : \text{Syn} \rightarrow \text{Syn},$$

an encoded transformation satisfies

$$\widehat{f}(\text{Enc}(s)) = \text{Enc}(f(s)).$$

Equivalently,

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

The continuous framework preserves precisely this commuting representation principle. For a topological space X , a continuous transformation

$$g : X \rightarrow X,$$

and a continuous encoding

$$\text{Enc} : X \rightarrow E,$$

the encoded transformation satisfies

$$\widehat{g}(\text{Enc}(x)) = \text{Enc}(g(x)),$$

or equivalently,

$$\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.$$

Thus the common structure is

objects	$\xrightarrow{\text{transformation}}$	objects
$\downarrow \text{encoding}$		$\downarrow \text{encoding}$
representations	$\xrightarrow{\text{encoded transformation}}$	representations.

In both settings, the representation is operational rather than merely identificatory: transformations of the represented objects correspond to transformations of their encodings.

10.2 What Is Generalized

The generalization occurs along several dimensions.

First, the source domain is no longer restricted to finite formal syntax. It may be a topological space

$$X$$

carrying continuous transformations.

Second, the target representation need not be a subset of the natural numbers. It may instead be a topological representation space

$$E$$

chosen to preserve or expose structure relevant to the transformation system.

Third, the transformation family may carry additional algebraic structure. In particular, group actions

$$\alpha : G \times X \rightarrow X$$

may be transported to encoded actions

$$\hat{\alpha} : G \times \text{Enc}(X) \rightarrow \text{Enc}(X),$$

preserving identity, composition, inverses, and orbit structure.

Fourth, the continuous setting requires a distinction absent from the purely topological transport problem: continuity does not imply computability. Executability therefore requires specified effective representations and computable realizers.

The resulting hierarchy may be summarized as

encoding $\longrightarrow$ equivariant transport $\longrightarrow$ structural preservation $\longrightarrow$ effective transport.

The first three levels are mathematical representation properties. The final level introduces the additional computational conditions required for executable symmetry.

10.3 What Is Not Being Claimed

The terminology introduced here should not be interpreted as asserting that arbitrary continuous systems satisfy analogues of Gödel's incompleteness theorems.

In particular, this paper does not claim that:

- (i) topological spaces possess Gödel sentences merely because they admit continuous encodings;
- (ii) continuous group actions exhibit incompleteness in the technical logical sense;
- (iii) real-valued or geometric representations are themselves arithmetizations of formal syntax;
- (iv) continuity provides a substitute for recursive or computable structure; or
- (v) faithful transport by conjugation constitutes a new incompleteness phenomenon.

Classical Gödel numbering has a specific logical role: it permits formal syntax and syntactic relations to be represented within arithmetic [1]. That arithmetization is a crucial ingredient in self-reference and incompleteness arguments.

The present construction extracts a different component of the same general architecture: the representation of objects together with transformations acting on their representations. It then

asks how that component behaves when the source system is continuous and when the encoded transformations are required to be executable.

Accordingly, the relationship to classical Gödel encoding is one of structural inheritance rather than theorem-level equivalence.

Remark 10.1. The commuting relation

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f$$

is considerably more general than classical Gödel numbering. Its appearance here should therefore not be regarded as uniquely Gödelian. What motivates the terminology is the route by which the framework is obtained: beginning with executable Gödel encoding, isolating its operational representation principle, and extending that principle from discrete formal syntax to continuous transformation systems.

11 Formalization and Verification

The preceding development is stated mathematically rather than as a completed machine-checked formalization. Nevertheless, the framework has a natural decomposition into definitions and proof obligations suitable for implementation in a proof assistant such as Lean 4 [3].

This section identifies that formalization boundary. No theorem in this paper is claimed to have been machine-checked unless accompanied by a corresponding verified implementation.

11.1 Formal Objects

A formalization of the core theory requires representations of the following objects:

$$X, \quad E, \quad \text{Enc} : X \rightarrow E, \quad \text{Dec} : \text{Enc}(X) \rightarrow X,$$

together with a family of transformations

$$f : X \rightarrow X.$$

At the topological level, the principal hypotheses are that Enc is a homeomorphism onto its image and that the source transformations are continuous.

A proof-assistant development may represent the encoded state space either explicitly as the image

$$\text{Enc}(X) \subseteq E$$

or as a subtype whose elements consist of an element of E together with evidence that it lies in the image of Enc .

The fundamental inverse laws are

$$\text{Dec}(\text{Enc}(x)) = x$$

for every $x \in X$, and

$$\text{Enc}(\text{Dec}(z)) = z$$

for every $z \in \text{Enc}(X)$.

These identities supply the rewriting principles used throughout the transport proofs.

11.2 Representation of Transformations

Given a source transformation

$$f : X \rightarrow X,$$

its encoded transformation is defined by

$$\widehat{f} := \text{Enc} \circ f \circ \text{Dec}.$$

The first formal verification target is the commuting theorem

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

The second is compositionality:

$$\boxed{\widehat{f \circ g} = \widehat{f} \circ \widehat{g}.}$$

The identity and inverse laws then become

$$\widehat{\text{id}_X} = \text{id}_{\text{Enc}(X)}$$

and, when f is invertible,

$$\widehat{f^{-1}} = \widehat{f}^{-1}.$$

These results require only the inverse laws for Enc and Dec and are therefore natural initial targets for mechanization.

For a group action

$$\alpha : G \times X \rightarrow X,$$

the encoded action is

$$\widehat{\alpha}(g, z) := \text{Enc}(\alpha(g, \text{Dec}(z))).$$

The corresponding formal proof obligations are

$$\widehat{\alpha}(e_G, z) = z$$

and

$$\widehat{\alpha}(g, \widehat{\alpha}(h, z)) = \widehat{\alpha}(gh, z).$$

Thus the algebraic portion of the theory can be formalized independently of the later computability layer.

11.3 Machine-Checked Invariants

The invariant-transport theorem provides another direct formalization target.

Suppose

$$I : X \rightarrow Y$$

satisfies

$$I(g(x)) = I(x).$$

Define

$$\widehat{I} := I \circ \text{Dec}.$$

The desired theorem is

$$\boxed{\widehat{I}(\widehat{g}(z)) = \widehat{I}(z).}$$

Likewise, for a group action, one may formalize orbit preservation:

$$G \cdot \text{Enc}(x) = \text{Enc}(G \cdot x).$$

These statements are mathematically proved in the present paper. A future machine-checked development would establish the same results inside the proof assistant and thereby verify that the definitions, typing conditions, and proof dependencies have been represented correctly.

Accordingly, the term “machine-checked” in this section denotes a formalization objective rather than a completed result of the present manuscript.

11.4 Executable Examples

The examples of Section 9 provide natural test cases for a future formal implementation.

For planar rotation, the central encoded operation is

$$\widehat{R}_\theta([\phi]) = [\phi + \theta].$$

A formal development should verify the commuting relation

$$\widehat{R}_\theta \circ \text{Enc} = \text{Enc} \circ R_\theta$$

and the composition law

$$\widehat{R}_\theta \circ \widehat{R}_\psi = \widehat{R}_{\theta+\psi}.$$

For the finite-dimensional linear example,

$$\text{Enc}(x) = Px$$

and

$$\widehat{A} = PAP^{-1}.$$

The corresponding formal target is

$$\widehat{A}\widehat{B} = \widehat{AB}.$$

The computability layer requires additional infrastructure. A full formalization would need explicit represented spaces, realizers, and the theorem that computable realizers are closed under composition. The Effective Transport theorem could then be represented formally as the implication

$$\boxed{\text{Enc, Dec, } f \text{ computable} \implies \text{Enc} \circ f \circ \text{Dec computable.}}$$

This separates two verification stages:

$$\boxed{\text{structural verification} \longrightarrow \text{effective verification.}}$$

The first verifies the algebraic and topological transport laws. The second verifies that the transported transformations admit executable realizers under explicitly chosen representations.

Remark 11.1. A proof assistant is particularly useful here because the distinction between structural existence and effective realization is reflected in different formal hypotheses. Continuity, invertibility, and equivariance do not by themselves provide a computability theorem. Effective representations and realizers must enter explicitly before executability can be formally derived.

12 Related Work

The framework developed in this paper lies at the intersection of several established areas: Gödel numbering and arithmetization of syntax, computable analysis and represented spaces, transformation groups and equivariance, and formal verification. The purpose of this section is to locate the present construction relative to these areas rather than to claim that its individual mathematical ingredients are new.

12.1 Gödel Numbering and Arithmetization

Classical Gödel numbering assigns natural-number codes to the symbols, formulas, and derivations of a formal system [1]. Its essential role is not merely to label syntactic objects, but to permit syntactic properties and operations to be represented arithmetically. Operations such as formation, substitution, and proof checking can thereby be reflected at the level of numerical codes.

The present framework begins from this operational feature of Gödel encoding, as also emphasized in the executable syntactic architecture of [4]. If

$$f : \text{Syn} \rightarrow \text{Syn}$$

is a syntactic transformation, its encoded counterpart satisfies

$$\widehat{f}(\text{Enc}(s)) = \text{Enc}(f(s)).$$

The continuous theory isolates this commuting relation and asks how an analogous representation architecture behaves when the source objects and transformations carry topological rather than merely discrete structure.

This use of the term “Gödel encoding” should therefore be understood historically and structurally. The general commuting relation

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f$$

is not specific to Gödel numbering, and the present paper does not claim otherwise.

12.2 Computable Analysis and Represented Spaces

Computable analysis addresses computation on objects such as real numbers, continuous functions, and other non-discrete mathematical structures [2, 5]. A standard framework is provided by represented spaces, in which an abstract object is accessed computationally through a name, commonly represented by an element of Baire space.

A map between represented spaces is computable when it possesses a computable realizer operating on such names. Admissibility and representation equivalence are developed in the established theory of represented spaces and qcb-spaces [6]. This distinction is essential for the present theory because continuity alone does not imply computability.

Section 8 therefore adopts represented spaces as the effective layer of continuous Gödel encoding. The Effective Transport theorem is an application of the closure of computable maps under composition: when

$$\text{Enc}, \quad f, \quad \text{Dec}$$

are computable relative to the chosen representations, the transported transformation

$$\widehat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is computable as well.

The present framework differs in emphasis from general computable analysis, rather than extending its general computability theorems. Its central object is a transformation system together with an encoding that is required simultaneously to preserve the source transformation semantics and to support effective execution under specified target constraints. The present results organize these obligations and illustrate their boundaries; they do not characterize all such constrained representations.

12.3 Transformation Groups, Conjugacy, and Equivariance

The transport

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

is mathematically a conjugation whenever Enc is an isomorphism between the relevant spaces. Preservation of identity, composition, inverses, and group actions under such transport is therefore standard structural mathematics.

Likewise, the commuting condition

$$\hat{f} \circ \text{Enc} = \text{Enc} \circ f$$

is an equivariance condition. Equivariant maps and group actions are standard tools for expressing the preservation of symmetry under a change of representation.

The present contribution is consequently not a new theory of conjugacy or group actions. Rather, these structures provide the mathematical language in which the operational principle extracted from executable encoding can be stated precisely.

12.4 Representation and Executability

A central distinction of the present framework is between passive representation and executable representation. A homeomorphism

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

automatically transports continuous transformations by conjugation. That fact alone supplies no new computational capability.

The executable problem begins when the target representation is specified independently. One may, for example, prescribe a data representation, a set of admissible primitive operations, an approximation discipline, or another computational constraint and then ask whether the transported transformation system remains realizable within that representation.

This viewpoint connects the structural transport problem with computable analysis while retaining the transformation-preservation condition inherited from the motivating Gödel-encoding setting.

12.5 Formal Verification

Proof assistants provide a complementary level of verification. The algebraic and topological statements developed here can be formalized as explicit proof obligations concerning encoding, decoding, composition, group actions, and invariant preservation.

A formal development would distinguish these structural proofs from the additional computability layer. In particular, machine verification of

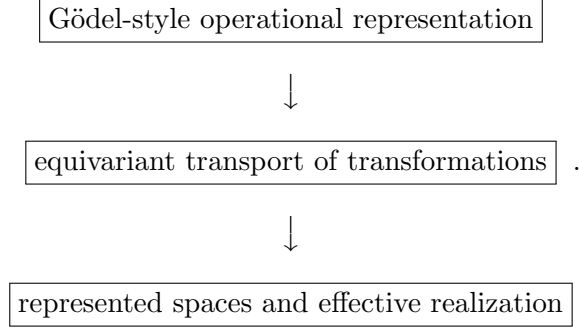
$$\hat{f} \circ \text{Enc} = \text{Enc} \circ f$$

does not by itself establish that $\widehat{f}$ has a computable realizer. The effective representations and corresponding computability hypotheses must also be formalized.

The present manuscript develops the mathematical framework and identifies these formalization targets, but does not claim a completed machine-checked implementation.

12.6 Position of the Present Framework

The relationship among these areas can be summarized as



Continuous Gödel encoding is proposed as a framework joining these three levels. Its distinctive question is not whether transformations can be transported across an isomorphic representation—which is automatic—but whether they can be transported into an independently constrained representation while retaining both their structural semantics and an effective realization.

13 Discussion

The preceding construction separates three questions that can easily be conflated: whether a transformation system can be represented in another space, whether its structural properties survive that representation, and whether the resulting representation supports effective execution. Continuous Gödel encoding is intended to make these three levels explicit.

13.1 From Executable Syntax to Executable Symmetry

The motivating pattern begins with executable representations of formal syntax. In the discrete setting, an encoding

$$\text{Enc} : \text{Syn} \rightarrow \mathbb{N}$$

permits a syntactic transformation

$$f : \text{Syn} \rightarrow \text{Syn}$$

to be represented by an operation on codes satisfying

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.$$

The present theory retains this operational relation while replacing the discrete syntactic domain by a topological transformation system. For

$$g : X \rightarrow X,$$

the corresponding relation becomes

$$\widehat{g} \circ \text{Enc} = \text{Enc} \circ g.$$

When the transformations arise from a group action, the encoded system preserves the identity, composition, inverses, orbits, and invariants of the source action. This yields a representation of symmetry on the encoded state space, but does not alone produce new algorithmic power; the effective transport and its failure under a noncomputable encoding are distinguished in Section 8.

The additional step from encoded symmetry to *executable* symmetry is not supplied by topology. It requires effective representations of the relevant spaces and computable realizers of the encoding, decoding, and transformation procedures.

The conceptual progression is therefore

$\begin{array}{ll} \text{operational encoding} & \longrightarrow \text{continuous transport} \\ \text{encoded symmetry} & \longrightarrow \text{effective realization} \end{array}$

This progression is the principal structural connection between executable Gödel-style encoding and the continuous framework developed here.

13.2 Limits of the Construction

Several limitations are essential to the interpretation of the results.

First, faithful transport through a homeomorphism is mathematically automatic. If

$$\text{Enc} : X \rightarrow X_{\text{Enc}}$$

is a homeomorphism, then

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

exists for every continuous transformation $f : X \rightarrow X$. The mere existence of $\hat{f}$ is therefore not a substantive computational result.

Second, continuity does not imply computability. A continuous transformation may fail to possess a computable realizer relative to the chosen representations. Claims of executability must therefore state the effective representations being used.

Third, topological faithfulness does not imply preservation of every geometric structure. Metrics, smooth structures, measures, symplectic forms, and other additional data require corresponding preservation hypotheses.

Fourth, effective conjugation can itself become formally automatic when the target computational representation is constructed entirely by transport from the source. The stronger problem is to begin with an independently constrained target representation and determine whether the desired transformations remain executable within it.

Finally, the term “Gödel” in continuous Gödel encoding refers to the operational representation architecture motivating the construction. No continuous analogue of the incompleteness theorems has been established or claimed.

13.3 Implications for Semantic Runtime Theory

The framework suggests a general architecture for runtimes whose states and transformations possess both semantic and executable representations.

Let X denote a semantic state space and let

$$\mathcal{F}$$

be a family of admissible state transformations. An encoded runtime uses a representation

$$\text{Enc} : X \rightarrow E$$

together with executable transformations

$$\widehat{\mathcal{F}}$$

such that

$$\widehat{f}(\text{Enc}(x)) = \text{Enc}(f(x))$$

for each admissible transformation f .

This relation supplies a correctness condition linking execution in the representation space to transformation in the semantic space. If the commuting condition is verified, an encoded execution step has a specified semantic interpretation.

For a sequence

$$f_1, f_2, \dots, f_n,$$

compositionality gives

$$f_n \circ \dots \circ f_1 = \widehat{f_n} \circ \dots \circ \widehat{f_1}.$$

Consequently, correctness can be stated compositionally across a sequence of encoded transformations.

This observation does not by itself produce a verified runtime. A concrete runtime must additionally specify its state representation, admissible operations, error model where approximation is involved, and the mechanism by which the required correctness properties are established.

Nevertheless, the framework identifies a reusable separation of concerns:

semantic state	$\leftrightarrow$	encoded state	$\leftrightarrow$	executable transformation.
----------------	-------------------	---------------	-------------------	----------------------------

The commuting condition then serves as the bridge between semantic meaning and represented execution.

13.4 Open Problems

The present framework leaves several mathematically distinct problems open.

Independently constrained representations. Given a transformation system $(X, \mathcal{F})$ and a target representation class $\mathcal{C}$, determine whether there exists an encoding

$$\text{Enc} : X \rightarrow E, \quad E \in \mathcal{C},$$

for which every required transformation admits an effective encoded realization. This is stronger than transport through an arbitrary homeomorphic copy of X .

Minimal executable representations. Among representations supporting a prescribed transformation family, one may ask whether there are representations minimizing description complexity, primitive operations, approximation cost, or another specified computational resource.

Approximate equivariance. Exact commutation requires

$$\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.$$

Numerical systems may instead satisfy an error-bounded relation such as

$$d_E(\widehat{f}(\text{Enc}(x)), \text{Enc}(f(x))) \leq \varepsilon.$$

A systematic theory should determine how such errors behave under composition and whether invariant information remains stable.

Smooth and geometric transport. For manifolds and Lie group actions, one may strengthen the framework by requiring diffeomorphic encodings or preservation of specified geometric structures. The interaction between such geometric constraints and effective representation remains to be developed.

Infinite-dimensional systems. Function spaces and other infinite-dimensional state spaces provide a substantially richer test of the represented-space formulation. In these settings, the choice of representation becomes especially important for computability.

Formal verification. A machine-checked implementation should formalize the structural transport theorems separately from the effective layer and verify the hypotheses required for executable examples.

Representation discovery. The deepest prospective problem is not merely to verify a proposed encoding but to determine how useful encodings are found. Given objects, transformations, invariants, and computational constraints, one may ask for procedures that identify representation spaces in which the required structure becomes both transparent and executable.

This last problem reverses the direction of the present development. Instead of beginning with an encoding and studying what it preserves, one begins with required structure and asks which representation makes that structure executable.

14 Conclusion

This paper developed a continuous extension of the operational representation principle underlying executable Gödel encoding. Rather than treating an encoding solely as an assignment of codes to objects, the framework considers objects together with the transformations acting on them and requires those transformations to commute with representation:

$$\boxed{\widehat{f} \circ \text{Enc} = \text{Enc} \circ f.}$$

For a topological space X and a faithful continuous encoding

$$\text{Enc} : X \rightarrow E,$$

continuous transformations of X are transported to the encoded state space

$$X_{\text{Enc}} = \text{Enc}(X)$$

by

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}.$$

This transport preserves composition, identity, and, where present, inverse transformations. For transformation families closed under composition, encoding transport yields an isomorphic encoded transformation system. For group actions, it preserves the action laws, orbit structure, and transported invariants.

These structural results alone do not establish executability. Transport by a homeomorphism is mathematically automatic, whereas effective execution requires additional computational structure. Using represented spaces, the paper therefore distinguished continuous transport from effective transport. When the encoding, decoding, and source transformation possess computable realizers, their composition

$$\hat{f} = \text{Enc} \circ f \circ \text{Dec}$$

also possesses a computable realizer. This provides a sufficient condition for executable continuous transformation.

The resulting framework separates three levels:

$$\boxed{\text{representation} \longrightarrow \text{structural preservation} \longrightarrow \text{effective realization.}}$$

Worked constructions illustrated these levels for one-parameter transformation groups, planar rotation, and finite-dimensional linear actions. In the rotation example, the independently specified angular representation reduces encoded rotation to addition modulo 2π , providing a concrete instance in which continuous symmetry admits an explicit effective realization.

The term *continuous Gödel encoding* refers to the extension of the operational representation architecture of Gödel-style encoding, not to a generalization of Gödel’s incompleteness theorems. The commuting representation principle itself belongs to the broader mathematics of conjugacy and equivariance. The additional question posed here is whether transformations can be represented within specified target constraints while retaining both their structural semantics and effective executability.

This leads to a broader representation problem:

$$\boxed{\text{Given a transformation system and computational constraints,} \\ \text{which representation makes its structure executable?}}$$

The present theory supplies the basic definitions and transport results needed to formulate that question precisely. Its further development requires independently constrained representations, error-aware approximate transport, richer geometric structures, and machine-checked implementations of the effective framework.

References

- [1] K. Gödel, “Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I,” *Monatshefte für Mathematik und Physik*, vol. 38, pp. 173–198, 1931. <https://doi.org/10.1007/BF01700692>.
- [2] K. Weihrauch, *Computable Analysis: An Introduction*. Berlin: Springer, 2000.
- [3] L. de Moura and S. Ullrich, “The Lean 4 Theorem Prover and Programming Language,” in *Automated Deduction—CADE 28*, Lecture Notes in Computer Science, Springer, 2021, pp. 625–635. https://doi.org/10.1007/978-3-030-79876-5_37.

- [4] A. Diamond, *Executable Gödel Encodings: A Verified Runtime for Logical Syntax*, preprint, 2026.
- [5] V. Brattka, P. Hertling, and K. Weihrauch, “A Tutorial on Computable Analysis,” in *New Computational Paradigms: Changing Conceptions of What Is Computable*, S. B. Cooper, B. Löwe, and A. Sorbi, eds., Springer, 2008, pp. 425–491. https://doi.org/10.1007/978-0-387-68546-5_18.
- [6] M. Schröder, “Admissibly Represented Spaces and Qcb-Spaces,” *Handbook of Computability and Complexity in Analysis*, Springer, 2021, pp. 305–346. <https://arxiv.org/abs/2004.09450>.