International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2024): 6.623

Category Theory and Psychodynamics: Bridging the Structure of Programming with Human Behavior

Tanmay Mendhey

Email: *tanmaytammy1993[at]gmail.com*

Abstract: ***This paper explores the intersection of category theory in functional programming and psychodynamic theory, offering a unique perspective on how categorical structures can be seen as metaphors for understanding human behavior. Through the lens of category theory, key concepts such as monads, inductive types, and optics are reframed as analogous to psychoanalytic structures like mental states, drives, and the therapeutic process. Drawing upon both mathematical abstraction and psychodynamic theory, the study highlights how algebraic structures in programming- such as functors and monads - can be interpreted as mechanisms for translating and transforming internal states within the human psyche. The application of directed type theory, parametricity, and homotopy type theory to software development is paralleled with psychodynamic techniques for understanding and transforming the unconscious mind. By connecting these seemingly disparate fields, the paper argues that the rigorous abstraction found in category theory can provide valuable insights into the organization and transformation of mental processes akin to the therapeutic practices of psychoanalysis.***

Keywords: Category theory, Functional programming, Monads, Type theory, Parametricity, Abstraction.

1. Introduction

1.1 Category Theory in Functional Programming

Several decades of functional programming practice in simply, polymorphically, and dependently typed programming languages have made it abundantly clear that there is virtue in identifying algebraic and categorical structures in program designs, as these abstractions enable code reuse by relying on abstract libraries for algebra and category theory.

While the design of Haskell, its libraries, and tutorials can create the impression that Type, the category of types and functions, is the *only* category we should care about, it is by now clear that this is absolutely not the case. To only scratch the surface of this fact:

- Pure functional programming languages tend to use mon-ads**red (**redFootnote about algebraic effects?**)** as a mathematical device for allowing non-purity [1], where specific monads only allow for specific forms of side-effects.
- **Monad morphisms** are then affect reinterpretations, ex-plaining the effects allowed by one monad in terms of the effects allowed by another. When monads are combined using monad transformers, then monad morphisms may arise from **morphisms of monad transformers**. When monads (or transformers) are indexed by other structures
- (e.g., the writer monad Writer $W = W \times \llcorner\ \lrcorner$, which allows programs to log messages, is indexed by a monoid of messages W), then monad morphisms arise from **morphisms between such structures**.
- Inductive data types are now understood as initial alge-bras μF of some polynomial functor (a.k.a. container functor) F [2], i.e. μF is the initial object in the **category of F-algebras**. The data needed to define a function by recursion on inductive data exactly constitutes an F-algebra A, with the motive being the carrier and each constructor clause being an algebraic operation.
- The function f: $\mu F \rightarrow A$ thus defined is the unique algebra morphism from the initial object μF, while the β-rules for μF exactly ensure that it is indeed an algebra morphism. Dually, data types are final coalgebras.

Optics such as lenses and traversals are bidirectional data accessors that can, e.g., read and update fields of record types (lenses) or entries of traversable functors (traversals). red (redCite nLab?) Optics themselves have identities and can be composed, **thus forming categories**, but they also have slick parametric encodings called Van Laarhoven optics red (redCite VL? Cite Haskell library?) and profunctor opticsred (redCite some paper? Cite Haskell library?), the **correctness of which relies on parametric/natural quantification over categories of (pro)functors with extra structure**.

1.2 Propagation and Preservation for Free

Beyond library support for category theory, three strains of research seek to provide *language* support for (adaptations of) category theory: directed type theory, parametricity, and homotopy type theory (HoTT).

1) *Directed Type Theory:* In Directed Type Theory (DirTT) [3], [4], [5], [6], [7], all types A are regarded as categories and typically come equipped natively with an indexed type $\mathrm{Hom}_A(a, b)$ of morphisms from $a : A$ to $b : A$. Very naively, we could hope for these morphisms to be preserved by *all* functions, meaning that all functions would be covariant functors. However, as anyone who has ever touched category theory knows, there are many interesting operations that are not a function at all. For this reason, all systems for DirTT hitherto developed (to my knowledge), feature one or more of the following internal features to escape this unreasonable requirement:
   - Non-transportive (Most related work speaks of *covariant* types instead of transportive types, but we prefer to reserve the word 'covariant' for the covariant modality) type families, i.e. type families $\Gamma, x : A \vdash T$ type where a morphism φ: $\mathrm{Hom}_A(a, b)$ does *not* give rise to a transport function $\varphi_* : T[a/x] \rightarrow T[b/x]$.
   - A non-directed-univalent universe U of non-transportive types, where a morphism of types $P : \mathrm{Hom}_U(A, B)$ does not correspond to a function $f : A \rightarrow B$ but behaves

more like a profunctor or even just a relation between A and B [4], [6], [7]. Thus, functions to U can preserve native morphisms without being functorial in the sense of producing functions.
- Modal annotations on function types, which express whether functions are co- or contravariant, both, or neither [8], [3], [9], [5], [7].

Progress in DirTT has historically been arduous, so the subject is currently in a much less advanced state than parametricity and HoTT for several reasons:

a) *Variance:* First, early on in the development of a directed type system, one is typically forced to make a choice on how to deal with variance: one can either use modal annotations on variables in the context and consider only transportive types with forward transport w.r.t. this modal context, or one can consider various forms of fibrancy w. r. t. an unannotated context. E.g., in 1-category theory:
- Grothendieck opfibrations (transportive types) provide forward transport,
- Grothendieck fibrations (optransportive types) provide backward transport,
- isofibrations provide only transport along isomorphisms,
- bifibrations provide an adjoint pair of tranpsort functions along every morphism.

We could, of course, have both in a single system and prove, e.g., that op fibrations over C correspond to fibrations over C^{op}, but most existing work commits to one approach, creating some divergence in the literature on directed type theory.

b) *Dimension stratification:* Secondly, I believe that the advancement of directed type theory has suffered from the idea that 'in dependent type theory, types are just terms, and universes are just big types.' This idea is, of course, tremendously helpful in sorting out the structure of judgments, contexts, and substitutions for dependent type theory, but it is at odds with a central fact in higher category theory, namely that the collection (universe) of (n, r)-categories is an $(n + 1, r + 1)$-category. We can, of course, truncate the dimension of this collection so as to contain it in a bigger version of itself, and regard
- The universe of $(0, 0)$-categories (i.e. sets) as a $(0, 0)$-category, thus cleanly modeling type theory with a universe in sets,
- The universe of $(\infty, 0)$-categories (i.e. ∞-groupoids) as an $(\infty, 0)$-category, as is done in HoTT,

or we can try to study the remarkable fixpoint of (∞, ∞)-categories.

In practice, all prior work on directed type theory (as well as the current) has regarded types as either finite-dimensional categories ($n < \infty$) or as being directed up to a finite dimension ($r < \infty$), and while a directed universe of sets (discrete fibrations/types) is seen in several papers [3], [4], [6], [7], universes with richer structures than the types they classify do not seem to have been explored yet at higher dimensions.

This has been particularly limiting in modally annotated systems for one-dimensional categories. The literature shows a tendency to couple the variance of dependent functions and their codomain in a way that seems detrimental to usability. For example, Licata and Harper [3] only allow the context $\Gamma, orange!60!black\oplus \mid x :^{white!70!black} A$ (Γ extended with a covariant variable x of type A) to be formed when the type A is a type w.r.t. Γ (i.e. $\Gamma \vdash A$ type), whereas the context $\Gamma, orange!60!black\ominus \mid x :^{white!70!black} A$ (x is a contravariant variable) can only be formed when A is type w.r.t. Γ (i.e. $\Gamma^{op} \vdash A$ type). The consequence is that we can only form a function type $\Gamma \vdash (orange!60!black \ominus \mid x :^{white!70!black} A) \to B$ type (dependent or not) of *contravariant* functions, since A needs to be for the function type to be, and then we can only have contravariant variables of type A.

Conversely, North [5] allows either context only to be formed when A is transportive w.r.t. Γ (i.e. $\Gamma \vdash A$ type). Both model contexts as categories, types as functors to **Cat**, and extended contexts as Grothendieck constructions, but the contravariant extended context is modeled differently:

$$[\![\Gamma, orange!60!black\ominus \mid x :^{white!70!black} A]\!] = \left(\int_{[\![\Gamma]\!]^{op}} [\![A]\!]\right)^{op}, \tag{1}$$

$$[3], \tag{2}$$

$$[\![\Gamma, orange!60!black\ominus \mid x :^{white!70!black} A]\!] = \int_{[\![\Gamma]\!]} \mathsf{Op} \circ [\![A]\!], \tag{3}$$

$$[5]. \tag{4}$$

i.e., Licata and Harper take the opposite of the total space of A, whereas North takes the opposite fibrewise. Neumann [10] provides the maximally configurable context extension that can be modeled as a Grothendieck construction: his context $\Gamma^u \triangleright^v x : A^w$ is parametrized by three bits: (u) whether or not we take the opposite of the preceding context, (v) whether A is transportive or optransportive w.r.t. Γ^u, and (w) whether or not we take the opposite of A fibrewise. However, once the Grothendieck construction is formed, we have a single category that we can only take the opposite of as a whole, toggling all three bits at once.

I argue that this is an artifact of disregarding the fact that Cat is more than just a category. Indeed, consider the universal directed type family, $X : \mathsf{Cat} \vdash X\,\mathsf{type}$. Because we already know what the type family is, the bits u and v in Neumann's context extension are coupled: X is transportive w.r.t. Cat and optransportive w.r.t. Cat^{op}. Still, this leaves us with two bits of freedom, which can only be toggled together by taking the opposite of the entire thing after we form the extended context. Concretely, the objects of the extended context will be pointed categories, and the morphisms will consist of a functor between the categories and a 'heterogeneous arrow' along it, and we have to commit to whether we want these heterogeneous arrows to point the same way as the functor, or the opposite way. We can call these laxly and oplaxly point-preserving functors.

However, if we were to model contexts as 2-categories and types as 2-functors to Cat, then it becomes clear that the direction of the arrows *within* a fiber of the type $\Gamma \vdash T\,\mathsf{type}$, is coupled to the direction of the 2-cells in Γ. Indeed, the 2-functor $\mathsf{Op}\colon \mathsf{Cat}^{\mathsf{co}} \to \mathsf{Cat}$ is covariant at level 1, but contravariant at level 2, so the fibrewise opposite type $\Gamma^{\mathsf{co}} \vdash T^{\mathsf{op}}\,\mathsf{type}$ is well-typed only in the 2-opposite context. This reveals that the Grothendieck construction forces us to collapse two things that naturally live at different dimensions.

Rather than choosing along which functors we consider heterogeneous arrows, we can acknowledge that they exist along functors pointing both ways and indeed along *zigzags* in Cat, i.e., chains of functors pointing alternating ways. Even more generally, we can consider heterogeneous arrows along profunctors $\mathsf{P} : \mathsf{C} \sim \mathsf{D}$, i.e. functors $\mathsf{P} : \mathsf{C}^{\mathsf{op}} \times \mathsf{D} \to \mathsf{Set}$. Since both functors $F : \mathsf{C} \to \mathsf{D}$ and $G : \mathsf{D} \to \mathsf{C}$ produce profunctors $\mathsf{Hom}_{\mathsf{D}}(F\llcorner\ \lrcorner, \llcorner\ \lrcorner), \mathsf{Hom}_{\mathsf{D}}(\llcorner\ \lrcorner, G\llcorner\ \lrcorner) : \mathsf{C} \sim \mathsf{D}$ and profunctors can be composed using the coend, zigzags to produce profunctors.

This encourages us to regard contexts not just as 2- categories but rather as *categories equipped with pro-arrows*, also more briefly called (pro-arrow) equipment, which are double categories with extra structure. In particular, Cat is equipment whose arrows are functors, whose pro-arrows are profunctors, and whose squares are heterogeneous natural transformations or, equivalently, profunctor morphisms. We then interpret types $\Gamma \vdash T\,\mathsf{type}$ as equipment functors $\mathcal{J}T) : \mathcal{J}\Gamma) \to \mathsf{Cat}$ and remark that the direction of both homogeneous $\to$ Set. However, and heterogeneous *arrows* in T is tied to the direction of pro- arrows in Γ: indeed, the functor $\mathsf{Op}\colon \mathsf{Cat}^{\mathsf{co}} \to \mathsf{Cat}$ is covariant on arrows but contravariant on pro-arrows.

A sensible and algebraically relevant semantics for the extended context Γ, *orange*!60!*black*μ 1 x :$^{white!70!black}$ T (for a certain modality μ) is now as the pro-arrow equipment of pointed categories, whose:

- Objects are pointed categories,
- Morphisms are point-*preserving* functors,
- Pro-arrows are pointed profunctors (i.e. a pro-arrow (P, ψ) : (C, c) ~ (D, d) is a profunctors P : C ~ D with a designated heterogeneous morphism $\psi \in$ P(c, d)),
- Squares are point-preserving profunctor morphisms.

Each time, things are paired up whose orientation was coupled anyway: the direction of the mapping relation '→: Obj(C) × Obj(D) → *Set* of a functor F: C → D is inevitably related to the direction of F itself, while the direction of a heterogeneous arrow $\psi \in$ P (c, d) along a profunctor P: C ~ D is also inevitably related to the direction of the P. The end result is new equipment in which arrows and pro-arrows can still independently be flipped.

The modality μ will be the directed version of the *structural* modality in Degrees of Relatedness [11], by which algebraic structures depend on their structure (e.g., pointed categories on their designated point).

We can generalize the above construction to an arbitrary type $\Gamma \vdash T\,\mathsf{type}$, i.e. an arbitrary equipment functor $T : \Gamma \to \mathsf{Cat}$, and call this the **equipment of elements**. The Grothendieck construction of a functor $F : \mathsf{C} \to \mathsf{Cat}$ can be defined in terms of the equipment of elements as follows:

1) Freely equip C with pro-arrows (which is left adjoint to forgetting that Cat has pro-arrows) obtaining an equipment functor $\bar{F} : \mathsf{FPro}\,\mathcal{C} \to \mathsf{Cat}$,
2) Take the equipment of elements $\int_{\mathsf{FPro}\,\mathcal{C}} \bar{F}$,
3) Extract its category of pro-arrows,
4) Restrict to those pro-arrows whose underlying pro-arrow in FPro C actually arises from an arrow in C, which can be done by taking a pullback.

c) *Naturality:* Thirdly, it has been unclear how to deal with the notion of *naturality* in a type system. In category theory, naturality is only defined in specific situations, such as with increasing generality (Note that these concepts are flexible w.r.t. the number and variance of arguments, as each of the mentioned categories can again be a product of (opposites of) categories, and moreover every functor independently has the right to be constant w.r.t. one or more of the arguments.

- natural transformations $\overset{\mathsf{nat}}{\forall}\, x.F\,x \to G\,x$ between functors $F, G : \mathsf{C} \to \mathsf{D}$ [12],
- extra natural transformations $\overset{\mathsf{exnat}}{\forall}\, x, y, z : F\,x\,x\,z \to G\,y\,y\,z$ between functors $F : \mathcal{A} \times \mathcal{A}^{\mathsf{op}} \times \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{B} \times \mathcal{B}^{\mathsf{op}} \times \mathcal{C} \to \mathcal{D}$ [13],
- natural transformations $\overset{\mathsf{dinat}}{\forall}\, x.F\,x\,x \to G\,x\,x$ between functors $F, G : \mathcal{C} \times \mathcal{C}^{\mathsf{op}} \to \mathcal{D}$,

which can all be defined together as a special case of the end $\overset{\mathsf{end}}{\forall}\, x.T\,x\,x$ of a factor $T : \mathcal{C} \times \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$.

However, natural transformations are a rather restrictive concept, while extranatural and dinatural transformations do not always compose. (The type signature of an extra-natural transformation specifies, besides the domain and codomain functors, a string diagram connecting each argument slot of the domain (codomain) functor to either a slot of opposite variance of the domain (codomain) functor or a slot of the same variance of the codomain (domain) functor. Extranatural transformations can be composed insofar as the composition of their string diagrams does not create closed 'bubbles' [14]. Petric´ provides a similar criterion for natural transformations [15], which McCusker and Santamaria independently reprove using Petri-nets [16].) This is essential because, given a morphism $\xi : x \to x'$, natural transformations have a different notion of heterogeneous equality in their domain and their codomain: in the domain, elements (in the internal language sense, i.e., morphisms to) of Fxx and $Fx'x'$ are considered equal if they *come from* the same element in Fxx', whereas in the codomain, elements of Gxx and $Gx'x'$ are considered equal if they *yield* the same element in $Gx'x$. Strong natural transformations

address this by considering elements of $F\,xx$ and $F\,x'x'$ equal if they *yield* the same element in $F\,x'\,x$. However, if we nest contravariance a bit deeper, strong naturality too proves inadequate for a general typing discipline: given a functor F: C × C^{op} → D, the function

$$\text{twice}: \forall x.\ (F\,xx \longrightarrow F\,xx) \longrightarrow (F\,xx \longrightarrow F\,xx), \quad (7)$$
$$\varphi\ '\!\!\longrightarrow \text{twice}_x\ \ \varphi: = \varphi \circ \varphi. \quad (8)$$

is *not* strongly natural because the notion of heterogeneous equality used for the domain and the codomain *of the domain* (and of the codomain) is different?

This problem appears to remain unsolved as of yet. It seems that the only type-theoretically adequate way to define the naturality of a (possibly mixed-variant, possibly deeply-nested contravariant) operation $\forall x.T\ x$, is to define heterogeneous equality for T by induction on the construction of T, as happens in Reynolds' relational model of parametricity for type theory [17] (section I-B2), as well as in the general presheaf model of type theory [18, ch. 4] (??). In this paper, I take the viewpoint that the correct generalization of naturality to quantifications over arbitrary mixed-variant types *is* parametricity w.r.t. graph 'relations' of morphisms. As such, we will build a system for parametric *and* directed type theory, where all types are natively equipped with notions of relations *and* morphisms, modeled in presheaf categories. Related work: [19], [20]

d) *The Hom-type:* A notable focal point of the difficulties related to naturality is the question of how to correctly type and model the identity morphism id: ∀ (x: A). Hom$_A$(x, x). In particular, we would like to understand Hom$_A$ as an inductive type in three ways: (1) Hom$_A$(a, ⌞ ⌟) is the initial functor F: A ⟶ Type such that $F\ a$ is inhabited; (2) Hom$_A$(⌞ ⌟, b) is the initial functor G : A^{op} ⟶ Type such that Gb is inhabited; and (3) Hom$_A$(⌞ ⌟, ⌞ ⌟) is the initial functor H : $A^{op} \times A$ ⟶ Type such that the end $\overset{\text{end}}{\forall}\ (x : A).H\ x\ x$ is inhabited.

Lacking any form of natural quantification for non-groupoidal types and having coupled variance of types and terms, North [5] defines a Hom-type with an identity morphism id : (x : A^{core}) ⟶ Hom$_A$(x, x) that is natural only w.r.t. the core of A and eliminates according to properties (1) and (2). However, this leaves the Hom-type underspecified: indeed, North models it as the semantic Hom-set, but it can alternatively be modelled as the non-isomorphic functor

$$[\![A]\!]^{\mathsf{op}} \times [\![A]\!] \to \mathsf{Set} : (a, b) \mapsto \overset{\mathsf{coend}}{\exists}\ (z : [\![A]\!]^{\mathsf{core}}).\mathrm{Hom}_{[\![A]\!]}(a, z) \times \mathrm{Hom}_{[\![A]\!]}(z, b),$$

which interprets an internal morphism from a to b as a semantic two-step morphism via an intermediate object Z that is also stored up to isomorphism.

DirTT is less advanced. Several reasons:

Hom-type: Bifibrations (did you mean two-sided?), naturality, parametricity. Mention treatment of modal inductive types. Interval

Parametricity: Old Stuff
We consider three strains of research that all seek to unburden users of proof assistants: parametricity, homotopy type theory (HoTT), and directed type theory (DirTT).

Parametricity HoTT Directed TT Everything has a graph: HoTT < DirTT < parametricity Computational content: para- metricity < HoTT and DirTT Functoriality: DirTT < HoTT and parametricity

2. Literature Review

Category theory has long played a significant role in functional programming, particularly in understanding algebraic structures that facilitate abstraction and code reuse. Research in type theory, including simply typed, polymorphic, and de- pendent types, has explored how *monads, functors, and optics* streamline software development. Existing works demonstrate the effectiveness of categorical structures in reasoning about *side effects, data transformations, and program composition*, with a heavy reliance on *Haskell's monadic abstractions*.

In parallel, psychodynamic theory has been a cornerstone of psychology, investigating how unconscious processes shape human behavior. Researchers in *computational cognitive science* have sought to model cognitive functions mathematically, often using *Bayesian networks, symbolic AI, and artificial neural networks* to describe thought processes. Despite advancements in computational models of the mind, relatively little work has explored the application of *category theory to psychology*. Some studies have examined *topos theory* in cognitive science and the potential use of *algebraic structures for reasoning about neural computations*, but no significant research directly links *psychodynamic processes* to categorical mechanisms such as *monads, functors, and type-theoretic abstractions*. This paper aims to fill that gap by establishing an interdisciplinary bridge between these fields.

3. Psychodynamic Concepts and Their Mathematical Analogies

A significant insight of this paper is that *mental states* *and transformations* can be framed using category-theoretic principles. Within category theory, objects can be understood as *mental states*, while morphisms represent *psychological transitions or cognitive restructuring*. Monads, which encapsulate computations within a structured pipeline, are useful in modeling *internal mental processes* such as information suppression, memory recall, and affect regulation.

One potential analogy is the *Monad of Suppression*, wherein a monadic structure encapsulates unconscious defense mechanisms, preventing certain thoughts from being evaluated, much like how monads in functional programming encapsulate computational side effects. Similarly, functors can be seen as *mechanisms of learning and experience transformation*, mapping one cognitive structure to another, akin to how therapy modifies deeply held beliefs.

Optics, such as lenses and traversals, offer another rich analogy by representing *cognitive focus and self-reflection*. Lenses provide a *bidirectional means of accessing and modifying parts of a mental framework*, which mirrors *therapeutic introspection*, while traversals model how individuals navigate between multiple interpretations of experience. These correspondences suggest that *category-theoretic structures can provide a new way of understanding cognitive change* in psychodynamic terms.

4. Comparison with Other Computational Cognitive Theories

Traditional computational models of cognition have relied heavily on *probabilistic reasoning, symbolic AI, and connectionist architectures*. Bayesian networks, for instance, offer a powerful framework for modeling decision-making and belief updating. However, they struggle to *express the hierarchical and structured transformations* inherent in psychodynamic processes.

Deep learning and neural networks attempt to capture cognitive processes using *gradient-based optimization*, but they often *lack interpretability* and a high-level algebraic structure. By contrast, a *category-theoretic approach introduces structured transformations between mental states*, making it a compelling alternative to existing models. Unlike Bayesian inference, which focuses on probability distributions over mental states, *category theory provides a formal structure to capture abstract relationships between states and transformations*. Compared to connectionist approaches, categorical models offer a level of *rigor and abstraction* that makes them well- suited for explaining *mental schemas, cognitive restructuring, and psychodynamic shifts*.

5. Case Studies and Applications

A concrete application of category theory to psychology lies in *therapeutic change and learning processes*. In *Cognitive Behavioral Therapy (CBT)*, individuals undergo structured interventions that modify their cognitive schemas. This can be seen as a functor that *maps initial dysfunctional thought patterns to healthier cognitive structures* while preserving key relationships within the mental system.

Another potential application is in *AI-driven psychoanalysis*, where category theory can provide formal models for how *emotional states evolve in structured ways over time*. AI models informed by these structures could allow for *better prediction and understanding of emotional responses* in natural language processing systems, particularly in chatbot therapy or AI-powered counseling services. This categorical framework could also help formalize the *dynamics of trauma processing, behavioral reinforcement, and learning cycles*.

6. Experimental and Practical Implications

The integration of category theory into computational psychology and AI could lead to multiple innovations. One promising direction is the *development of AI models for emotional intelligence* that employ categorical structures to formally represent *state transitions in cognitive and emotional processing*. Such AI systems could better predict and classify mental states based on structured transformations, making them valuable for *automated therapy, personalized coaching, and behavioral predictions*.

Another practical implication is in *the development of computational frameworks for psychodynamic modeling*, where *monads, functors, and type-theoretic approaches could be implemented in AI-based cognitive architecture*. This could lead to advances in *psychological modeling software, intelligent tutoring systems, and AI-driven mental health diagnostics*.

7. Future Research Directions

Further work is needed to refine this framework and validate its applicability. One research avenue is to explore *higher-category structures* to model meta-cognitive processes and *self-referential transformations* in the psyche. Another promising direction is the incorporation of *Homotopy Type Theory (HoTT)* as a means of formalizing *continuous and structured changes in cognitive models*.

Additionally, an important step would be to translate these theoretical concepts into *practical, AI-driven applications* that assist in psychological modeling, mental health interventions, and cognitive architecture in artificial intelligence. As interdisciplinary interest in formal cognitive models grows, category theory may play a critical role in *shaping the future of AI- powered behavioral sciences*.

8. Conclusion

This paper has proposed a novel conceptual bridge between *category theory and psychodynamic psychology*, arguing that the algebraic structures underlying programming languages can provide valuable insights into *mental state transformations and psychological processes*. By framing mental processes using *monads, functors, optics, and directed type theory*, we open the door to new perspectives in *computational psychology, AI- assisted therapy, and cognitive modeling*.

Future research will be necessary to refine these analogies, apply them to *real-world psychological frameworks*, and implement them in *computational AI systems*. With further interdisciplinary collaboration, this category-theoretic approach may significantly advance both *computer science and psychology* in the quest to understand and simulate human cognition.

References

[1] E. Moggi, "Computational lambda-calculus and monads," in *4th annual symposium on logic in computer science*. IEEE Press, 1989, pp. 14–23.

[2] M. Abbott, T. Altenkirch, and N. Ghani, "Containers: Constructing strictly positive types," *Theoretical Computer Science*, vol. 342, no. 1, pp. 3–27, 2005, applied Semantics: Selected Topics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397505003373

[3] D. R. Licata and R. Harper, "2-dimensional directed type theory," *Electr. Notes Theor. Comput. Sci.*, vol. 276, pp. 263–289, 2011. [Online]. Available: https://doi.org/10.1016/j.entcs.2011.09.026

[4] E. Riehl and M. Shulman, "A type theory for synthetic ∞-categories," *ArXiv e-prints*, May 2017.

[5] P. R. North, "Towards a directed homotopy type theory," in *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4-7, 2019*, ser. Electronic Notes in Theoretical Computer Science, B. Ko¨nig, Ed., vol. 347. Elsevier, 2019, pp. 223–239. [Online]. Available: https://doi.org/10.1016/j.entcs.2019.09.012

[6] M. Z. Weaver and D. R. Licata, "A constructive model of directed univalence in bicubical sets," in *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbru¨cken, Germany, July 8-11, 2020*, H. Hermanns, L. Zhang, N. Kobayashi, and D. Miller, Eds. ACM, 2020, pp. 915–928. [Online]. Available: https://doi.org/10.1145/3373718.3394794

[7] D. Gratzer, J. Weinberger, and U. Buchholtz, "Directed univalence in simplicial homotopy type theory," *CoRR*, vol. abs/2407.09146, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.09146

[8] A. Abel, "Polarised subtyping for sized types," *Mathematical Structures in Computer Science*, vol. 18, no. 5, pp. 797–822, 2008. [Online]. Available: https://doi.org/10.1017/S0960129508006853

[9] A. Nuyts, "Towards a directed homotopy type theory based on 4 kinds of variance," Master's thesis, KU Leuven, Belgium, 2015. [Online].Available: https://anuyts.github.io/files/mathesis.pdf

[10] J. Neumann, "Towards modal sogats," in *EuroProofNet Work Group 6 Meeting*, 2024. [Online]. Available: https://europroofnet.github.io/ wg6-leuven/programme#neumann

[11] Nuyts and D. Devriese, "Degrees of relatedness: A unified framework for parametricity, irrelevance, ad hoc polymorphism, intersections, unions and algebra in dependent type theory," in *Logic in Computer Science (LICS) 2018, Oxford, UK, July 09-12, 2018*, 2018, pp. 779–788. [Online]. Available: https://lirias.kuleuven.be/retrieve/510956

[12] S. Eilenberg and S. MacLane, "General theory of natural equivalences," *Transactions of the American Mathematical Society*, vol. 58, no. 0, p. 231–294, 1945. [Online]. Available: http://dx.doi.org/10.1090/S0002-9947-1945-0013131-6

[13] G. Kelly, "Tensor products in categories," *Journal of Algebra*, vol. 2, no. 1, pp. 15–37, 1965. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021869365900220

[14] S. Eilenberg and G. Kelly, "A generalization of the functorial calculus," *Journal of Algebra*, vol. 3, no. 3, pp. 366–375, 1966. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021869366900068

[15] Z. Petric´, "G-dinaturality," *Annals of Pure and Applied Logic*, vol. 122, no. 1, pp. 131–173, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168007203000034

[16] G. McCusker and A. Santamaria, "Composing dinatural transformations: Towards a calculus of substitution," *Journal of Pure and Applied Algebra*, vol. 225, no. 10, p. 106689, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022404921000256

[17] J. C. Reynolds, "Types, abstraction and parametric polymorphism." In *IFIP Congress*, 1983, pp. 513–523.

[18] M. Hofmann, *Syntax and Semantics of Dependent Types*. Cambridge University Press, 1997, pp. 79–130.

[19] M. S. New and D. R. Licata, "A formal logic for formal category theory," in *Foundations of Software Science and Computation Structures 26th International Conference, FoSSaCS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, ser. Lecture Notes in Computer Science, O. Kupferman and P. Sobocinski, Eds., vol. 13992. Springer, 2023, pp. 113–134. [Online]. Available: https://doi.org/10.1007/978-3-031-30829-16

[20] A. Laretto, F. Loregian, and N. Veltri, "Directed equality with dinaturality," 2024. [Online]. Available: https://arxiv.org/abs/2409.10237