What underlies dual-process cognition? Adjoint and representable functors

Steven Phillips (steve@ni.aist.go.jp)
Mathematical Neuroinformatics Group, Human Informatics Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8566 JAPAN

Abstract

Despite a general recognition that there are two styles of thinking: fast, reflexive and relatively effortless (Type 1) versus slow, reflective and effortful (Type 2), dual-process theories of cognition remain controversial, in particular, for their vagueness. To address this lack of formal precision, we take a mathematical category theory approach towards understanding what underlies the relationship between dual cognitive processes. From our category theory perspective, we show that distinguishing features of Type 1 versus Type 2 processes are exhibited via adjoint and representable functors. These results suggest that category theory provides a useful formal framework for developing dual-process theories of cognition.

Keywords: dual-process; Type 1; Type 2; category theory; category; functor; natural transformation; adjoint

Introduction

Intuitively, at least, human cognition involves two starkly different styles of thinking: one style appears to be fast, reflexive and relatively effortless; the other slow, reflective and effortful (Evans, 2003; Kahneman, 2011). The former is called Type 1 and the latter Type 2 (Evans & Stanovich, 2013). Distinguishing features are listed in Table 1.

ID	Type 1	Type 2
D1	no working memory	working memory
D2	autonomous	decoupled, simulation
F1	fast	slow
F2	high capacity	low capacity
F3	parallel	serial
F4	unconscious	conscious
F5	biased responses	normative responses
F6	contextualized	abstract
F7	automatic	controlled
F8	associative	rule-based
F9	experience-based	consequence-based
F10	ability-independent	ability-dependent
F11	evolved early	evolved late
F12	animal	human
F13	implicit	explicit
F14	basic emotions	complex emotions

Table 1: Definitions/Features (D/F) of Type 1/2 processes (adapted from Evans & Stanovich, 2013, Table 1).

Dual-process theories of such styles of thinking are controversial. Evans and Stanovich (2013) reviewed five general criticisms of dual-process theories, which are summarized in Table 2. Foremost is the acknowledgement that dual-process theories are multitudinous and vaguely defined. The Intervention model attempts to address this problem (see Evans & Stanovich, 2013, for the model and responses to the five criticisms). According to this model, cognition defaults to Type 1 processes that can be intervened upon by Type 2 processes in response to task demands. However, this model is still far from the kinds of formal, computational explanations that prevail in cognitive science.

ID	Criticism of dual-process theories
C1	Theories are multitudinous and definitions vague
C2	Type 1/2 distinctions do not reliably align
C3	Cognitive styles vary continuously, not discretely
C4	Single-process theories provide better explanations
C5	Evidence for dual-processing is unconvincing

Table 2: Five criticisms of dual-process theories (Evans & Stanovich, 2013).

We propose a category theory (Mac Lane, 1998) approach to understanding dual-process cognition. The attraction of this approach is the precise formalization of the relationships between cognitive processes that have some of the distinctive features of Type 1 versus Type 2 cognition. An intuitive basis for the category theory that supports this approach is given in the next section. Examples are sketched in the section that follows, and this approach is discussed with regard to the five criticisms in the final section. Technical support appears in the appendix.

Categories and functors

Here, we provide an intuitive overview of the supporting theory given in the appendix (see, e.g., Awodey, 2010; Mac Lane, 1998, for introductions to category theory). To aid intuition, formal concepts are interpreted in terms of the more familiar notion of an analogy between source and target domains. Our main interest is in adjoint and representable functors, which depend on the concepts of category, functor and natural transformation. So, we proceed in that order. Some conceptual correspondences are summarized in Table 3.

Category theory	Cognitive science
category	(sub)system, source/target
functor	construction, analogy
natural transformation	comparison (of analogies)
adjoint (functor)	inverse/obverse, Type 1 $\leftrightarrow$ 2
representable (functor)	representation

Table 3: Corresponding categorical and cognitive concepts.

A proportional analogy serves to bootstrap basic intuitions about categories, functors and natural transformations: *Predecessor is to successor as one is to two*; *as Monday is to Tuesday*. Analogy is generally recognized as a map of entities in a source domain to entities in a target domain that preserves their relations (Gentner, 1983): e.g., next(predecessor, successor) in the order domain maps to next(one, two) in the number domain; likewise, next(Monday, Tuesday) in the days-of-the-week domain. In this context, we can interpret:

- a *category* (definition 1) as a source/target domain, which consists of a collection of *objects* (e.g., predecessor, successor), a collection of *morphisms* (e.g., next) between objects, and a *composition operation* for combining morphisms (e.g., next *of* next);
- a *functor* (definition 3) as an analogy (e.g., days-of-the-week) from one category (source) to another category (target) that "preserves structure" (e.g., *next of next* in the order domain maps to *next of next* in the number domain); and
- a *natural transformation* (definition 4) as a comparison of analogies (e.g., next number and next day) so that the result of applying a relation to a transformation is the same as the transformation of the result of applying a relation (e.g. the day after the first day is the same as the second day).

Natural transformations also appear to be like analogies, but the appearance belies an important distinction: natural transformations are maps between functors, whereas functors are maps between categories: 2nd-order contra 1st-order analogy.

With these concepts we can proceed to adjunction, regarded as the centrepiece of *ordinary* category theory (Mac Lane, 1998), and the related notion of representable functor. Our motivation is the observation that Type 1/2 processes are distinguished by counterposing features and immediacy (e.g., autonomous versus simulated). To illustrate, consider *approximation* and *precision* as conceptual inverses, though not actual inverses since approximation discards information: e.g., reals are approximated by integers, but their sets are not isomorphic. Yet, there is a "second-order isomorphism" between their orders affording comparisons with reals in terms of integers *without* loss of precision. We can interpret:

- an *adjunction* (definition 5) as an "inverse" (obverse) relation between two "anti-parallel" functors: e.g., one functor sends each real to its *ceiling* (approximation: e.g., $2.3 \mapsto 3$) and the other functor sends each integer to its corresponding real (precision: e.g., $3 \mapsto 3.0$); and
- a *representable functor* (definition 6) as a construction that can be represented, or simulated by another functor (e.g., comparisons with reals in terms of integers).

The integers, likewise reals, with the usual order form a category (example 1(a)). The two number systems are related by adjunctions (example 6(a, b)) and associated representable functors (example 7). Effectively, comparisons of reals to upper and lower (integer) bounds can be computed in terms of integers, thus avoiding a need for infinite precision. In a cognitive context, the infinite real-world is represented by finite resources. This example also shows how representable functors convey a second-order isomorphism even though a first-order isomorphism does *not* exist.

Categorical perspective on dual-processes

Type 1 and Type 2 processes are distinguished by counterposed features, suggesting adjoint and representable functors as the formal connection. We sketch cases centred on pairs of adjoint functors. In each case, we lead with an aspect of cognition that motivates a particular adjunction, which suggests related distinctions that follow,

Whole/part: diagonal-product functors

One commonly held distinction is whether cognitive processes operate on cognitive representations wholistically or componentially (whole/part). For example, *kicked the bucket* can be interpreted wholistically (idiomatically) as *died*, or componentially as an act of *kicking*. The first interpretation is characteristically associative and the second rule-based (F8).

This whole/part distinction is expressed by functors that pertain to a componential object called a *product*. Products are constructed by the *product functor*, which is right adjoint to the *diagonal functor* (example 6(c)). This case is a conceptual inverse in the sense of combining parts to form wholes in one direction (product functor) and contextualizing wholes as parts in the other direction (diagonal functor).

This adjoint situation was tested in a paired-associates task: subjects learned to associate pairs of letters to coloured shapes, where say first letters were associated with colours and second letters to shapes (Phillips, Takeda, & Sugimoto, 2016). In this situation, the dual processes are a map of letter pairs versus of pair of letter maps. Performance on generalization (novel) trials indicated associative versus rule-based processes, which correlated with conscious awareness (F4) of the underlying rule. Unaware subjects did not show correct responses to cues beyond the context of those seen in training trials: did not infer an abstract rule that extended to novel letter pairs (F6). Thus, the diagonal-product functor expresses three distinctive features of Type 1/2 processes: F4, F6 and F8.

Parallel/serial: product-exponential functors

A second common distinction is between parallel versus serial cognitive process (F3). Visual search, for example, is classically regarded as involving either a parallel process when the time to find a target item is independent of the number of non-targets, or a serial process when time increases linearly with the number of non-targets (Treisman & Gelade, 1980).

The parallel/serial distinction is expressed by the *product-exponential* adjunction (example 9). As applied to functions, this adjunction is called the *(un)curry* transform in computer science: e.g., $+(x,y) \Leftrightarrow +(x)(y)$. Here, the general advantage of parallel processing is speed, since multiple arguments are applied concurrently. Thus, the product-exponential adjoint expresses distinctions F1 and F3.

Automatic/controlled: free-forgetful functors

A third distinction pertains to automatic versus controlled processes (F7). For example, counting a small number of items, *subitizing*, is automatic whereas counting a large number of items is controlled by an incremental process.

The free-forgetful adjunction (example 10) expresses this distinction. Counting is modelled as a monoid: the natural numbers with addition, and zero as the identity element—counting is a serial process (F3) starting at zero and adding one until all items are counted. The free functor sends a set to the free monoid on that set, which affords the counting process. The forgetful functor sends a monoid to its underlying set, forgetting the monoid operation. So, the free functor constructs the control process, whereas the forgetful functor constructs the corresponding automatic process, a map that obviates the control steps, expressing the distinction between automatic and controlled processes (F7). The automatic process associates lists of items to their count, the controlled process steps through each list item (F8), hence the automatic process is fast (F1) and effectively parallel (F3) since the intermediate counting steps are obviated. Thus, the free-forgetful adjoint expresses distinctions F1, F3, F7 and F8.

Autonomous/simulated: representable functors

A fourth distinction is between autonomous versus simulated processes (D2). This distinction is regarded as definitive of Type 1 versus Type 2 processes, and is sometimes distinguished as on-line versus off-line processing (Halford, Wilson, Andrews, & Phillips, 2014). The central advantage of off-line processing is that one can assess the potential effects of an action without having to incur its consequences. That is the advantage of working with a representation, or mental model of the world, in place of the world itself.

Representable functors express this difference between autonomous and simulated processes. Every adjunction induces a pair of representable functors (remark 7). So, all the previous examples of adjoint functors involve representable functors. A representable functor does two things: (1) preserves the structure of the domain category in the form of sets and functions in the category **Set**, and (2) in such a way that the structure is represented (or, modelled) by a hom-functor (example 3). For adjoint functors between categories **C** and **D**, the domain of associated representable functors consists of pairs of objects and morphisms from **C** and **D**. The interaction between these categories in the form of an adjunction is represented by sets and functions in **Set** (D2). Forgetful functors are representable functors (example 12), so the forgetful functor in the previous example is representable. This functor is represented by the natural numbers monoid and its generator (remark 9). This representation makes explicit the relationship between each natural number (F13). The natural numbers are represented by a process that generates them. Thus, representable functors express distinctions D2 and F13.

This categorical method of representation is generalized further, by the Yoneda lemma (lemma 1) and functor (definition 7), in terms of morphisms between objects. Suppose, e.g., a world as a 2-dimensional space (example 13). In this context, the Yoneda functor affords a coherent representation of this world with respect to given reference points A and B (diagram 3), such as a cognitive agent's binocular sensors. Moreover, isomorphisms are preserved (remark 11), in which case, one view can be used to calibrate the other.

Discussion

In this section, we discuss our category theory approach in terms of the five general criticisms levelled against dual-process theories, which were summarized in Table 2, and then provide some overall perspective.

C1 Dual-process theories are criticized for their vagueness. Category theory provides a formally precise foundation for the relationship between cognitive dual-processes, in terms of adjoint and representable functors. Moreover, although we have presented several different pairs of adjoint and representable functors, all such situations derive from the same general form. Thus, our categorical approach addresses this criticism with precision and parsimony.

C2 Another criticism is that distinguishing features do not align with Type 1 and Type 2 processes. Evans and Stanovich (2013) countered that only D1 and D2 are supposed to be definitive of Type 1 and Type 2 processes, wheres F1–14 are only supposed to correlate with process type. The analogous situation, here, is that all representable functors derived from adjoints have the same form, thus they all align with D2. The correlated features cluster with different adjoints. Thus, the categorical approach is consistent with the role of these features. Note, however, the categorical approach currently does not say anything about working memory (D1), and several other correlated distinctions (F2, F5, F9–12, and F14).

C3 To some extent, the categorical approach is neutral with regard to the criticism that dual-route theories imply discreteness, whereas the data suggest continuity of alternatives. That is because the nature of the alternative path via natural transformations depends on the nature of the functors being related. Natural transformations also apply to relations between continuous maps, and were originally introduced to address such situations. The categorical approach does not necessitate discreteness or continuity of process type. Even for discrete alternatives, decision criteria may lie on a continuous dimension, such as a cost/benefit trade-off (see Phillips, Takeda, & Sugimoto, 2016), giving the appearance of a continuity.

C4 Whether or not category theory provides a better explanation for dual-process cognition remains to be determined. However, the close relationship between adjoint functors, universal constructions, and systematicity points in favour of a categorical approach. Adjunctions are another kind of (categorical) universal construction, universal constructions were employed to explain systematicity (Phillips & Wilson, 2010), and one can argue that dual-process is a general systematic property of human cognition. Thus, category theory provides a better explanation in this regard.

C5 Evidence favouring dual-process over single-process theories has been given (see, e.g., Evans & Stanovich, 2013; Kahneman, 2011). Support for an adjoints approach was provided in an experiment designed to test the diagonal-product basis for dual-processes (Phillips et al., 2016). With enough parameters, a single-process model can be devised to fit the same data. The question is whether such parameters (assumptions) are *ad hoc*: serve only to fit rather than explain the data. This question relates to the previous response (**C4**), so one area of inquiry is data showing systematicity as a property in the context of dual-styles of thinking.

Perspective

The main advantage of a categorical approach is the precise formalization of the relationship between dual processes. This advantage is particularly important as dual-process theories are applied to cognitive development (Evans, 2011), since one needs to know how they are related. The use of adjoints is particularly relevant to cognitive science, which predominantly bases the notion of structure-sensitive processing on isomorphism. However, as illustrated, isomorphism is often too strict to be a useful criterion, hence the relevance of the often cited quote, "Adjoints and everywhere" (Mac Lane, 1998). An adjunction can be regarded as a kind of local (as opposed to global) isomorphism: bijections between hom-sets (remark 2), without requiring that the collections of objects (between the respective categories) be isomorphic. So, one can reasonably expect adjoint and representable functors to play a role in other distinctive forms of dual-processing, which we have not addressed here.

Nonetheless, there are three general issues to be addressed in further work. The first issue concerns the adjoint basis for dual-process. Adjunctions pertain to equalities between the alternative computational paths. Thus, the current approach does not address cases where the dual processes supply contradictory responses. One possibility is to use adjunctions in higher category theory where the paths are themselves related by isomorphisms, instead of equalities.

A second issue concerns the link between the theory and empirical data. Some work has been done in this direction. Experiments were designed around the diagonal-product adjunction to test the empirical basis for non-systematic versus systematic properties of cognition (Phillips et al., 2016), which one can see as an aspect of the associative/rule-based distinction (F8). A question for any dual-process theory is to explain why one process is executed over the other. Further empirical work suggested a kind of cost/benefit trade-off as an account of which route (Phillips, Takeda, & Sugimoto, 2017). The theoretical challenge for our approach is to provide a categorical basis for such trade-offs.

A third issue concerns computational mechanisms. Category theory also provides a rigorous foundation for computation, particularly recursion (see, e.g., Hinze & Wu, 2016). Thus, we expect that category theory will also afford computational methods for cognitive dual-processes.

As a final remark, note the expository style of this work as an illustration of our adjoint basis for dual-process cognition: an informal main text on one hand and a formal appendix on the other. The main text is an approximate, yet more readily accessible exposition with links to the latter more precise, yet densely written alternative. Ideally, space permitting, the appendix would also provide links back to the main text (cf. approximation and precision as adjoints). Bidirectional exploitation of such trade-offs, in general form, is seen as the quintessential advance of human cognition (Phillips, 2017).

Acknowledgements

This work was supported by a Japanese Society for the Promotion of Science Grant-in-aid (16KT0025).

References

Awodey, S. (2010). *Category theory* (2nd ed.). New York, NY: Oxford University Press.

Evans, J. S. B. T. (2003). In two minds: Dual-process accounts of reasoning. *Trends in Cognitive Sciences*, *7*(10), 454–459.

Evans, J. S. B. T. (2011). Dual-process theories of reasoning: Contemporary issues and developmental applications. *Developmental Review*, *31*, 86–102.

Evans, J. S. B. T., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: advancing the debate. *Perspectives on Psychological Science*, *8*(3), 223–241.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 47–59.

Halford, G. S., Wilson, W. H., Andrews, G., & Phillips, S. (2014). *Categorizing cognition: Toward conceptual coherence in the foundations of psychology*. MIT Press.

Hinze, R., & Wu, N. (2016). Unifying structured recursion schemes: an extended study. *Journal of Functional Programming*, *26*, e1.

Kahneman, D. (2011). *Thinking, fast and slow*. New York, NY: Farrar, Straus and Giroux.

Mac Lane, S. (1998). *Categories for the working mathematician* (2nd ed.). New York, NY: Springer.

Phillips, S. (2017). A general (category theory) principle for general intelligence: duality (adjointness). In T. Everitt, B. Goertzel, & A. Potapov (Eds.), *Artificial general intelligence* (Vol. 10414, pp. 57–66). Springer.

Phillips, S., Takeda, Y., & Sugimoto, F. (2016). Why are there failures of systematicity? The empirical costs and benefits of inducing universal constructions. *Frontiers in Psychology*, *7*, 1310.

Phillips, S., Takeda, Y., & Sugimoto, F. (2017). Dual-routes and the cost of determining least-cost. *Frontiers in Psychology*, *8*, 1943.

Phillips, S., & Wilson, W. H. (2010). Categorial compositionality: A category theory explanation for the systematicity of human cognition. *PLoS Computational Biology*, *6*(7), e1000858.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*(1), 97–113.

Appendix A: Category theory

Definition 1 (Category)**.** A *category* $\mathbf{C}$ consists of a collection of *objects*, $O(\mathbf{C}) = \{A, B, \dots\}$, a collection of *morphisms*, $\mathcal{M}(\mathbf{C}) = \{f, g, \dots\}$—a morphism written in full as $f : A \to B$ indicates object A as the *domain* and object B as the *codomain* of f—including for each object $A \in O(\mathbf{C})$ the *identity morphism* $1_A : A \to A$, and a *composition* operation, $\circ$, that sends each pair of *compatible* morphisms $f : A \to B$ and $g : B \to C$ (i.e. the codomain of f is the domain of g) to the *composite* morphism $g \circ f : A \to C$, that together satisfy the laws of:

- *identity*: $f \circ 1_A = f = 1_B \circ f$ for every $f \in \mathcal{M}(\mathbf{C})$, and
- *associativity*: $h \circ (g \circ f) = (h \circ g) \circ f$ for every triple of compatible morphisms $f, g, h \in \mathcal{M}(\mathbf{C})$.

Remark 1. A morphism $f : A \to B$ with an inverse g is called an *isomorphism*: $f \circ g = 1_B$ and $g \circ f = 1_A$. We write $A \cong B$.

Example 1 (Category)**.** The following are categories.

a The integers with the usual order relation, $(\mathbb{Z}, \leq)$, form a category that consists of objects $x \in \mathbb{Z}$ and morphisms $x \leq y$ (identities are $x \leq x$), with composition given by transitivity.

b The category **Set** has sets for objects, functions for morphisms, and function composition as the composition operation. Identity functions are the identity morphisms.

c The *opposite category* $\mathbf{C}^{\mathrm{op}}$ has $\mathbf{C}$-objects with morphisms reversed, i.e. $f : A \to B$ in $\mathbf{C}$ is $f : B \to A$ in $\mathbf{C}^{\mathrm{op}}$.

d *Product category* $\mathbf{C} \times \mathbf{D}$ has $\{(A, B) | A \in O(\mathbf{C}), B \in O(\mathbf{D})\}$ for its collection of objects, $\{(f, g) | f \in \mathcal{M}(\mathbf{C}), g \in \mathcal{M}(\mathbf{D})\}$ for its collection of morphisms, and pointwise composition.

Remark 2. $\mathrm{Hom}_{\mathbf{C}}(A, B)$ is the set of morphisms in category $\mathbf{C}$ with domain A and codomain B. $\mathbf{C}$ is elided when understood.

Definition 2 (Post/precomposition)**.** Given a morphism f,

- *postcomposition* with f is the operation $f^* : h \mapsto f \circ h$ and
- *precomposition* with f is the operation $f_* : h \mapsto h \circ f$.

Remark 3. Postcomposition and precomposition combine, for example, $f_* g^* : h \mapsto f_*(g \circ h) = g \circ h \circ f$.

Definition 3 (Functor)**.** A (*covariant*) *functor* is a "structure-preserving" map from a category $\mathbf{C}$ to a category $\mathbf{D}$, written $F : \mathbf{C} \to \mathbf{D}$, sending each object A and morphism $f : A \to B$ in $\mathbf{C}$ to the object $F(A)$ and the morphism $F(f) : F(A) \to F(B)$ in $\mathbf{D}$ (respectively) that satisfies the laws of:

- *identity*: $F(1_A) = 1_{F(A)}$ for every object $A \in O(\mathbf{C})$, and
- *compositionality*: $F(g \circ_{\mathbf{C}} f) = F(g) \circ_{\mathbf{D}} F(f)$ for every pair of compatible morphisms $f, g \in \mathcal{M}(\mathbf{C})$.

Example 2 (Functor)**.** The following are functors.

a $\mathit{Incl} : (\mathbb{Z}, \leq) \to (\mathbb{R}, \leq); x \mapsto x$, see example 1(a).

b $\mathit{Ceil} : (\mathbb{R}, \leq) \to (\mathbb{Z}, \leq); x \mapsto \lceil x \rceil$, e.g., $\lceil 2.4 \rceil = 3$.

c $\mathit{Floor} : (\mathbb{R}, \leq) \to (\mathbb{Z}, \leq); x \mapsto \lfloor x \rfloor$, e.g., $\lfloor 2.4 \rfloor = 2$.

d $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}; A \mapsto (A, A), f \mapsto (f, f)$.

e $\Pi : \mathbf{C} \times \mathbf{C} \to \mathbf{C}; (A, B) \mapsto A \times B, (f, g) \mapsto f \times g$.

Functors a, b and c preserve order: e.g., $x \leq y \Rightarrow \lceil x \rceil \leq \lceil y \rceil$.

Remark 4. $F : \mathbf{C}^{\mathrm{op}} \to \mathbf{D}$ is called a *contravariant functor*.

Example 3 (Hom-functor)**.** The following are hom-functors.

a The *covariant* hom-functor pertains to postcomposition: $\mathrm{Hom}(A, -) : \mathbf{C} \to \mathbf{Set}; X \mapsto \mathrm{Hom}(A, X), g \mapsto g^*$.

b The *contravariant* hom-functor pertains to precomposition: $\mathrm{Hom}(-, B) : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}; X \mapsto \mathrm{Hom}(X, B), f \mapsto f_*$.

c The *bivariate* hom-functor combines postcomposition and precomposition: $\mathrm{Hom}(-, -) : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{Set}; (A, B) \mapsto \mathrm{Hom}(A, B), (f, g) \mapsto f_* g^*$, see remark 3.

Example 4 (Bivariate hom-functor)**.** Two additional bivariate hom-functors are obtained by precomposing with functor

a $F : \mathbf{C} \to \mathbf{D}$ on the left argument: $\mathrm{Hom}(F-, -) : \mathbf{C}^{\mathrm{op}} \times \mathbf{D} \to \mathbf{Set}; (A, B) \mapsto \mathrm{Hom}(F(A), B), (F(f), g) \mapsto F(f)_* g^*$ and

b $G : \mathbf{D} \to \mathbf{C}$ on the right argument: $\mathrm{Hom}(-, G-) : \mathbf{C}^{\mathrm{op}} \times \mathbf{D} \to \mathbf{Set}; (A, B) \mapsto \mathrm{Hom}(A, G(B)), (f, G(g)) \mapsto f_* G(g)^*$.

Definition 4 (Natural transformation)**.** Let $F, G : \mathbf{C} \to \mathbf{D}$ be functors. A *natural transformation* $\eta : F \dot{\to} G$ is a family of $\mathbf{D}$-morphisms $\{\eta_A : F(A) \to G(A) | A \in O(\mathbf{C})\}$ such that $G(f) \circ \eta_A = \eta_B \circ F(f)$ for every morphism $f : A \to B$ in $\mathbf{C}$.

Remark 5. A *natural isomorphism* is a natural transformation where every η_A is an isomorphism, see remark 1.

Example 5 (Natural hom)**.** Hom-functors relate naturally.

a Covariantly, $\mathrm{Hom}(h, -) : \mathrm{Hom}(A, -) \dot{\to} \mathrm{Hom}(B, -)$.

b Contravariantly, $\mathrm{Hom}(-, h) : \mathrm{Hom}(-, A) \dot{\to} \mathrm{Hom}(-, B)$.

Remark 6. The functors $\mathbf{C} \to \mathbf{D}$ (objects) and their natural transformations (morphisms) form a category, denoted $\mathbf{D}^{\mathbf{C}}$.

Definition 5 (Adjunction)**.** An *adjunction* from category $\mathbf{C}$ to category $\mathbf{D}$ is a tuple, $(F, G, \eta, \varepsilon) : \mathbf{C} \rightharpoonup \mathbf{D}$, consisting of functors $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$, and natural transformations $\eta : 1_{\mathbf{C}} \dot{\to} G \circ F$ and $\varepsilon : F \circ G \dot{\to} 1_{\mathbf{D}}$ such that $G\varepsilon \circ \eta G = 1_G$ and $\varepsilon F \circ F\eta = 1_F$. F is the *left adjoint* of G (G is the *right adjoint* of F), denoted $F \dashv G$, and η (ε) is the *(co)unit*.

Example 6 (Adjunction)**.** The following are adjunctions.

a $\mathit{Ceil} \dashv \mathit{Incl}$ with unit $x \leq \lceil x \rceil$ and counit $y \leq y$.

b $\mathit{Incl} \dashv \mathit{Floor}$ with unit $x \leq x$ and counit $\lfloor y \rfloor \leq y$.

c $\Delta \dashv \Pi$ with unit $\langle 1,1\rangle : Z \to Z \times Z$ and counit $(\pi_1, \pi_2) : (A \times B, A \times B) \to (A, B)$. In **Set**, $\pi_1 : (a,b) \mapsto a$, etc.

Remark 7. Every adjunction induces a natural isomorphism: $\phi : \mathrm{Hom}(F-,-) \cong \mathrm{Hom}(-,G-) : \psi$ (Mac Lane, 1998), as indicated by the following *commutative diagram*:

$$\begin{array}{ccc} \mathrm{Hom}_{\mathbf{D}}(F(A),B) & \underset{\psi_{A,B}}{\overset{\phi_{A,B}}{\rightleftarrows}} & \mathrm{Hom}_{\mathbf{C}}(A,G(B)) \\ \downarrow{\scriptstyle \mathrm{Hom}(F(h),k)} & & \downarrow{\scriptstyle \mathrm{Hom}(h,G(k))} \\ \mathrm{Hom}_{\mathbf{D}}(F(A'),B') & \underset{\psi_{A',B'}}{\overset{\phi_{A',B'}}{\rightleftarrows}} & \mathrm{Hom}_{\mathbf{C}}(A',G(B')) , \end{array} \tag{1}$$

which states that $\mathrm{Hom}(F(h),k) = \psi_{A',B'} \circ \mathrm{Hom}(h,G(k)) \circ \phi_{A,B}$ and $\mathrm{Hom}(h,G(k)) = \phi_{A',B'} \circ \mathrm{Hom}(F(h),k) \circ \psi_{A,B}$.

Example 7 (Bounds)**.** Instantiating $F \dashv G$ as:

- *Ceil* $\dashv$ *Incl* yields $\lceil x \rceil \le y \Leftrightarrow x \le y$, and
- *Incl* $\dashv$ *Floor* yields $x \le y \Leftrightarrow \lfloor x \rfloor \le y$ (see example 2).

Example 8 (Diagonal-product)**.** The diagonal functor is left adjoint to the product functor ($\Delta \dashv \Pi$, example 6(c)), hence the bijection $\mathrm{Hom}((Z,Z),(A,B)) \cong \mathrm{Hom}(Z, A \times B)$. In **Set** with $Z = 1$, i.e. singleton set $\{*\}$, the bijection is between two sets of "points"—a point is a map $\overline{a} : 1 \to A; * \mapsto a$, where $a \in A$. Thus, diagram 1 specializes and simplifies to

$$\begin{array}{ccc} (A,B) & \underset{\psi}{\overset{\phi}{\rightleftarrows}} & A \times B \\ \downarrow{\scriptstyle (f,g)} & & \downarrow{\scriptstyle f \times g} \\ (A',B') & \underset{\psi'}{\overset{\phi'}{\rightleftarrows}} & A' \times B' , \end{array} \tag{2}$$

i.e. the equivalence between a map of pairs and a pair of maps.

Example 9 (Product-exponential)**.** The product functor $\Pi_B : \mathbf{C} \to \mathbf{C}; A \mapsto A \times B, f \mapsto f \times 1_B$ is left adjoint to the exponential functor $\Lambda^B : \mathbf{C} \to \mathbf{C}; C \mapsto C^B, f \mapsto f^B$. In **Set**, the object C^B is the *function space* $\{f : B \to C\}$, hence the bijection $\mathrm{Hom}(A \times B, C) \cong \mathrm{Hom}(A, C^B)$, which specializes to the equality: $f(a,b) = \tilde{f}(a)(b)$, where $\tilde{f} : a \mapsto f_a$ and $f_a : b \mapsto c$. This equivalence is called *curry-uncurry* in computer science.

Example 10 (Free-forgetful)**.** A *monoid* is a set together with a binary operation and an identity element, e.g., the naturals with addition and 0 as the identity, $(\mathbb{N}, +, 0)$. **Mon** is the category of monoids and monoid homomorphisms. The *free* functor $F : \mathbf{Set} \to \mathbf{Mon}; S \mapsto (S, \star, e)$ sends each set S to the free monoid on S, e.g., the free monoid on an alphabet A is the set of all "words" A^* with the concatenation operation $\cdot$ and e as the empty word. The *forgetful* functor $U : \mathbf{Mon} \to \mathbf{Set}; (M, \star, e) \mapsto M$ sends each monoid to its underlying set, forgetting the operation, which is right adjoint to the free functor, $F \dashv U$. In this situation, the correspondence between the monoid homomorphism $h : (A^*, \cdot, e) \to (\mathbb{N}, +, 0)$ and the function $h : A^* \to \mathbb{N}$ is word length.

Definition 6 (Representable functor)**.** A *representable functor* is a functor $F : \mathbf{C} \to \mathbf{Set}$ such that there exists a hom-functor $\mathrm{Hom}(A,-) : \mathbf{C} \to \mathbf{Set}$ naturally isomorphic to F: there exists a pair (A, ϕ) where $A \in O(\mathbf{C})$ and $\phi : \mathrm{Hom}(A,-) \cong F$.

Remark 8. Contravariantly, we require $\psi : \mathrm{Hom}(-,B) \cong F$.

Example 11 (Hom-functor)**.** Hom-functors are representable functors being isomorphic to themselves.

Example 12 (Forgetful)**.** Forgetful functor $U : \mathbf{Mon} \to \mathbf{Set}$ (see example 10) is a representable functor, being naturally isomorphic to the hom-functor $\mathrm{Hom}(F(1),-)$, where $F(1)$ is the free monoid on the singleton set $1 = \{*\}$. In example 10, $(\mathbb{N}, +, 0)$ is the free monoid on $\{1\}$.

Remark 9. The pair $(\mathbb{N}, 1)$ is called a *representation* of U, where 1 is the generator of the natural numbers.

Lemma 1 (Yoneda)**.** Given functor $F : \mathbf{C} \to \mathbf{Set}$ and object $A \in O(\mathbf{C})$, there is a bijection from the set of natural transformations $\mathrm{Nat}(\mathrm{Hom}(A,-),F)$ to the elements of $F(A)$ that is natural in F and A, written $y_{F,A} : \mathrm{Nat}(\mathrm{Hom}(A,-),F) \cong F(A)$.

Remark 10. Setting F to hom-functor $\mathrm{Hom}(B,-)$ yields the bijection $\mathrm{Nat}(\mathrm{Hom}(A,-),\mathrm{Hom}(B,-)) \cong \mathrm{Hom}_{\mathbf{C}}(B,A)$.

Definition 7 (Yoneda functor)**.** A *Yoneda functor* is a functor $\mathcal{Y} : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}^{\mathbf{C}}; A \mapsto \mathrm{Hom}(A,-), h \mapsto \mathrm{Hom}(h,-)$.

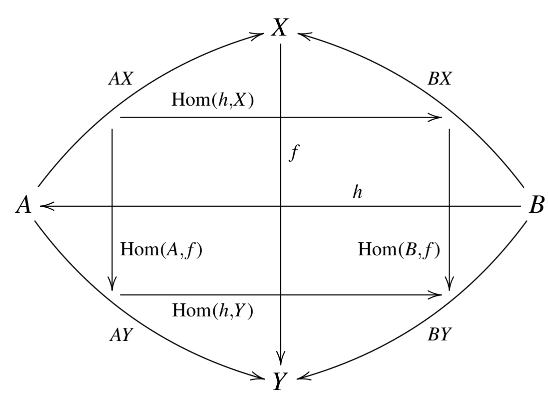
Remark 11. $\mathcal{Y}$ is an *embedding*: injective on objects and bijective on hom-sets (remark 10), so $A \cong B \Leftrightarrow \mathcal{Y}(A) \cong \mathcal{Y}(B)$.

Example 13 (Yoneda functor)**.** The coordinate space $\mathbb{R}^2$ is the category whose objects are points $(A_x, A_y) \in \mathbb{R}^2$, morphisms $AB : A \to B$ are translations $(B_x - A_x, B_y - A_y)$—identities are zero translations—and composition is addition. The Yoneda functor $\mathcal{Y} : \mathbb{R}^2 \to \mathbf{Set}$ sends:

- each object $A \in O(\mathbf{C})$ to hom-functor $\mathrm{Hom}(A,-) : X \mapsto \{AX\}, f \mapsto f+$, where f is an f-translation, and
- each morphism $h : B \to A$ to the natural transformation $\mathrm{Hom}(h,-)$, where h is an h-translation.

Objects/morphisms are indicated in the following diagram:

(3)



where the arrows between arrows are functions in **Set** and all other arrows are translations in $\mathbb{R}^2$. Conceptually, consider A and B as points of reference with regard to entities X and Y and their relationship f in the world, $\mathbb{R}^2$, and the commutative rectangle as a coherent view (representation) of that world.