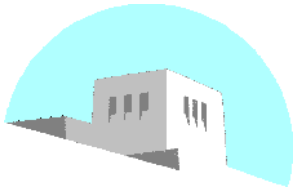
3-10-2010

Temporal Sequencing via Supertemplates

Michael Healy

Thomas Caudell

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING



SCHOOL OF ENGINEERING UNIVERSITY OF NEW MEXICO

Temporal Sequencing via Supertemplates

Michael J. Healy[1]
Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico 87131, USA
e-mail:mjhealy@ece.unm.edu

Thomas P. Caudell
Department of Electrical and Computer Engineering
and Department of Computer Science
University of New Mexico
Albuquerque, New Mexico 87131, USA
e-mail:tpc@ece.unm.edu

UNM Technical Report: EECE-TR-10-0001

Report Date: July 20, 2010

[1]This work was supported in part by the Defense Threat Reduction Agency (DTRA), United States Department of Defense, under Grant No. HDTRA1-08-1-0053.

Abstract

A category-theoretic account of neural network semantics has been used to characterize incremental concept representation in neural memory. It involves a category of concepts and concept morphisms together with categories of objects and morphisms representing the activity in connectionist structures at different stages of weight adaptation. Colimits express the more specialized concepts as combinations of abstract concepts along shared subconcept relationships specified in diagrams. This provides a mathematical model of concept blending, in which designated relationships among concepts are preserved in a combination. Structure-preserving mappings called functors from the concept to neural categories provide a mathematical model of incremental concept representation through stages of adaptation. The work reported here extends these ideas to express temporal sequences of events, such as episodic memories. This requires an extended notion of neural morphism and a design principle for diagrams involving concepts in a temporal sequence. This is tested in a new architecture that involves a notion of supertemplates, which are ART network templates extending over a multi-level ART hierarchy with an interposed temporal integrator network.

Keywords

ART, category, colimit, concept, connection path, diagram, episode, event, functor, morphism, neural, semantics, temporal sequence, theory

1 Introduction

In this paper, we apply a mathematical theory of concept representation and learning in neural networks to the derivation of a network that can learn and re-enact temporal sequences of events. We are using it to model episodic memory storage and recall, particularly in connection with the hippocampus and adjoining brain structures, and sequence encoding in general (some representative work in these areas can be found in [19, 1, 12, 14]). The newly-derived neural network is a combination of a previously-known architecture (the Fuzzy ART network, [2]), a neural network layer that performs temporal integration, and additional neural structure suggested by category theory. The resulting composite architecture generates a modified ART template which, instead of being internal to a single ART network, embraces a hierarchy of two ART networks; we call this a *supertemplate*.

1.1 A Categorical Semantic Theory

In previous work, we have proposed and tested a theoretical model of knowledge representation in neural networks, the categorical neural semantic theory (CNST) (although not by that name; see any of [10, 9, 11, 7]). The questions that can be addressed with the CNST include: What concepts express the meaning implicit in stimuli and/or autonomous, internally-generated signals that produce specific patterns of activity in a network? How do these concepts relate to each other and how are those relations represented in the network? Can we identify mechanisms by which, through connection-weight adaptation, new concept representations are formed in the network from existing concept representations together with incoming stimuli and internally-generated activity? Because it analyzes network activity and connection-weight adaptation in terms of concept representation, the CNST is a declarative semantic model. It associates the connectionist structure of a neural network at any stage of adaptation with a knowledge structure expressing facts and suppositions about possible worlds, or domains. This knowledge representation is based upon the network's experience with its stimulus generating environment, its initial structure, and any autonomous activity. Because it is mathematically based, the CNST is a *formal* declarative semantic model. The mathematics is that of category theory, the study of structures and structure-preserving relationships at all levels of abstraction. The CNST applies mathematical structures derived via category theory to express interrelated systems of concepts and structure-preserving mappings of category theory to formalize the incremental representation of these concept structures in neural structure and operation. This expresses the meaning implicit in the transformation of input stimuli to neural network output, learning through long-term adaptation, and retrieval from memory.

The CNST is one of three known category-theoretic models of neural structure and processing, each taking a unique approach [4, 6, 10]. The purpose of the CNST is to characterize the semantics of the significant states of processing in a network. The semantic analysis associates concepts with various sets of outputs for each node and concept relationships with bundles of network pathways between nodes. The concepts are descriptions of the stimuli that are associated with the output sets for the nodes. This is explained in detail in [10], and that which is essential to the present topic will be summarized here. Other references contain information about the theory and some initial applications of it; for example, see [8, 9, 20, 11, 7]. In the CNST, concepts are expressed as theories in a formal logic and their relationships as theory morphisms, which together consititute the objects and morphisms of a category. Many-to-one structure-preserving mappings called functors map this category to categories representing the structure and processing of neural networks at their various stages of adaptation in a given environment. In each functorial mapping, the many-to-one aspect of a functor "smears" or "compresses" the representation of the infinite number of concepts and relationships that are not represented explicitly in the network of interest (Fig. 1). Mappings called natural transformations are structure-preserving associations between functors; these model coherent processing among the functional regions of a neural network based upon their individual, functorial knowledge representations. Adaptation in the network is expressed as a transition from one neural category to the next, with a concomitant change to the mappings.

The CNST has recently been applied to model the memory storage and retrieval operations involved in processing temporal sequences of events, and in particular episodic memories. As mentioned in the first paragraph, the neural network we have developed to begin exploring temporal sequence learning comprises multilevel ART

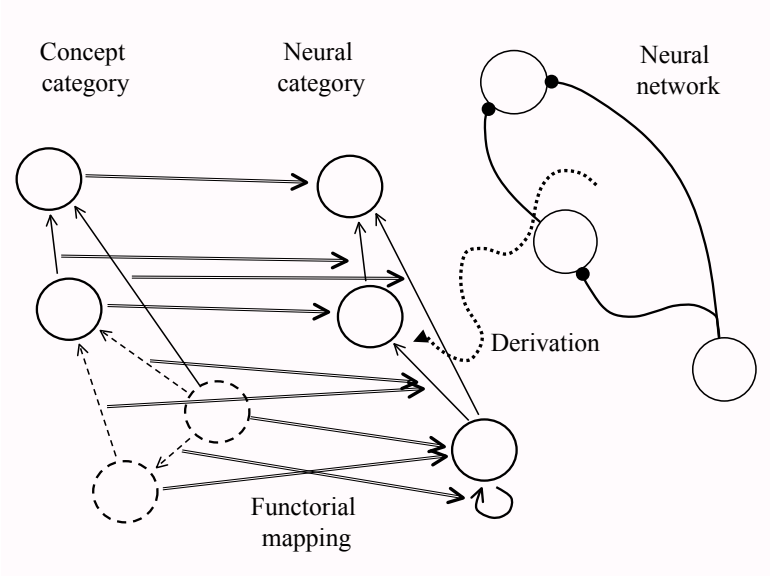


Figure 1: **Many-to-one functorial mapping.**

modules, a temporal integrator module, and the supertemplate connection structure suggested by the CNST.

Section 2 provides a brief review of the CNST. Section 3 applies the CNST to an anlysis of temporal sequencing in neural architectures. In Section 4, we present an initial neural architecture for temporal sequencing and discuss the neural category associated with the architecture, which leads to a modifies architecture incorporating the supertemplate connections. An initial experiment in temporal sequence learning and replay with the architecture, comparing results obtained with and without the supertemplate connections, is the subject of Section 5; graphical figures from the simulations are displayed in appendices A, B, and C. Section 6 is the conclusion.

2 Concept Representation: The CNST

Category theory (see for example [3, 13, 15, 17]) is based upon the notion of an *arrow*, or *morphism* (the two terms are used interchangeably) between two *objects* in a *category*. A morphism $f\!:\!a \longrightarrow b$ has a *domain* object a and a *codomain* object b, and serves as a directed relationship between a and b. The significance of this notion of arrow or morphism, and what distinguishes a category from a directed multigraph, is that a category has the property of *compositionality*. That is, in a category C, each pair of arrows $f : a \longrightarrow b$ and $g : b \longrightarrow c$ (where the codomain b of f is also the domain of g as indicated) has a *composition* arrow $g \circ f : a \longrightarrow c$ whose domain a is the domain of f and whose codomain c is the codomain of g. Composition is associative: For three arrows $f\!:\!a \longrightarrow b$, $g\!:\!b \longrightarrow c$ and $h\!:\!c \longrightarrow d$, the result of composing them is order-independent, with $h \circ (g \circ f) = (h \circ g) \circ f$. For each object a, there is an *identity morphism* $\mathtt{id}_a\!:\!a \longrightarrow a$ such that for any arrows $f\!:\!a \longrightarrow b$ and $g\!:\!b \longrightarrow a$, $\mathtt{id}_a \circ g = g$ and $f \circ \mathtt{id}_a = f$. A familiar example of a category is one called **Set**, which has sets as its objects, functions as its morphisms, and whose composition is just the composition of functions, $(g \circ f)(x) = g(f(x))$.

Key notions for the theoretical background of this paper are *commutative diagrams*, *initial and terminal objects*, and *limits* and *colimits*. A diagram is a collection of objects and morphisms of C. In a commutative diagram, any two morphisms with the same domain and codomain, where at least one of the morphisms is the composition of two or more diagram morphisms, are equal. An initial object, where one exists in C, is an object

i for which every object a of C is the codomain of a unique morphism $f: i \longrightarrow a$. A terminal object t has every object a of C as the domain of a unique morphism $f: a \longrightarrow t$. Limits will not be discussed here, and colimits will be introduced with an illustration showing why they are needed. In several papers including [10], we have shown how colimits model the learning of complex concepts through re-use of simpler concepts already represented in the connection-weight memory of a neural network. In [7] we have applied both limits and colimits to modify a neural network architecture (a version of ART) and thereby improve its performance in a class of applications.

A mapping between categories that preserves compositional structure, called a *functor*, formalizes the notion of transporting one type of structure, represented by a category, into another type of structure that can support a representation of the first structure. For categories C and D, a functor $F : C \longrightarrow D$ associates to each object a of C a unique image object $F(a)$ of D and to each morphism $f : a \longrightarrow b$ of C a unique morphism $F(f) : F(a) \longrightarrow F(b)$ of D. As mentioned in Section 1, a functor is in general a many-to-one mapping, where many objects of C can map to a single object in D, and similarly for morphisms (as long as domains and codomains are properly mapped). But a functor is more than simply a mapping. In order for F to be a functor, it must be the case that for each composition $g \circ_C f$ in C, the following holds: $F(g \circ_C f) = F(g) \circ_D F(f)$, where $\circ_C$ and $\circ_D$ denote the respective compositions in C and D. Along with this property, it is required that for each object a of C, $F(\mathtt{id}_a) = \mathtt{id}_{F(a)}$. It follows that F maps commutative diagrams of C to commutative diagrams in D. This means that any structural constraints expressed in C are translated into D. Finally, as also mentioned, natural transformations unify different functorial mappings. They will not be discussed here, but it is important to mention them because they fill important roles in the semantic theory and can be used in multiregion network design. For example, they can express the fusion of the separate concept representations contained in the processing regions of different sensor modalities in a multisensor neural network (for a brief description, see [10]).

2.1 A Category for Neural Network Semantics

Symbolically-expressed concepts can be regarded as descriptions of possible worlds or domains, internal models which the network constructs from stimuli, internally synthesized activity patterns, combinations of the two, or time-varying sequences of these. We refer to all these forms as "input", whether stimulus or synthesized. The stimuli are inputs sampled from the environment via the network's sensor(s). Depending upon the network's ability to synthesize activity, the worlds may or may not exclusively model the stimulus environment. Is the network activity consistent solely with events occurring in the environment or are there systematic deviations that signify autonomous operation? In any case, with an appropriate modelling capability in the hands of the analyst, the possible worlds can be expressed as a system of concepts both abstract and specific. The concepts depend upon the neural activity and connection-weight adaptation, or learning, that creates the representations from the foregoing forms of input, for they determine the network's future response. The capability of a given neural network to represent a given level of conceptual knowledge, of course, lies in its connection structure and operational rules for stimulus response and weight adaptation. With the CNST as a modelling tool, we analyze the knowledge representation capability in a given network, or, alternatively, design for this capability in either a new or modified network. In the scheme of the CNST, a network learns concept representations by re-using the already-represented concepts in many ways in combination with its inputs to "discover" concepts not yet represented in the connection-weight array of the network.

We express knowledge mathematically as a category **Concept**, whose objects are concepts and whose morphisms are similar to "sub-concept" relationships. This is familiar to categorical logicians as a category of formal logic theories ([3, 5, 16]; and see [10]). A theory morphism $s: T \longrightarrow T'$ (if one exists with T as domain and T' as codomain) is a replacement of the symbols of T by those of T' that transforms the axioms of T into either axioms or theorems of T'. The composition of morphisms by composing symbol substitutions is straightforward. This provides a mathematical expression of the compositional, hierarchical (and parallel, distributed) structure of knowledge. For example (see Fig. 2), a theory of polygons, such as triangles, includes a theory of geometry that expresses points, lines, a notion of "betweenness" (for points that lie "between" two points on a line, or "between" two rays emanting from a point), and angles. These, in turn, are expressed in simpler theories, some

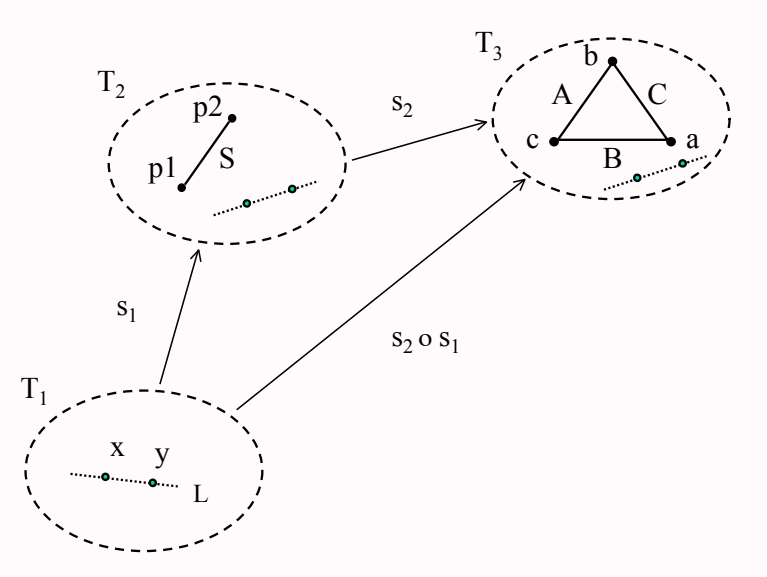


Figure 2: **Composition of morphisms in the Concept category.**

of them interrelated. The inclusion of points and lines in a theory of rays is a morphism, as is the inclusion of a theory of rays in a theory of angles (all points "between" two rays). Points, lines, rays, and angles are all included in a theory of, say, triangles, and the morphisms describing the relationships among these theories constitute a hierarchy of knowledge beginning with "first principles" (the properties of points and lines). For brevity, we refer the reader to any of [10, 9, 7] for a detailed discussion of theories and their morphisms. For our purposes, the concepts the theories express will be illustrated pictorially, and the morphisms likewise.

Fig. 2 illustrates a simple composition of morphisms in **Concept**. The morphism s_1 is a mapping of the symbols and syntax of the simple geometry theory T_1, which expresses only points and lines, into the theory of T_2, which expresses line segments bounded by point pairs and highlights a particular line segment labelled S. Expressing T_2 requires additional information to that from T_1, including a notion of "betweenness" for points on a line. The key requirement on a theory morphism is that the mapping be truth-preserving: That is, after substituting the names of symbols in T_2 (or, more generally, symbol strings which are well-formed formulas) into the syntax of T_1, all resulting statements are valid in T_2—that is, they are provable from the axioms of T_2. The morphism s_1 thereby explains the dependence of theory T_2 on theory T_1, and does so with mathematical rigor. The morphism s_2 likewise maps T_2 into the theory of a triangle (T_3), in particular mapping S to one of the three sides of the triangle. The composition morphism $s_2 \circ s_1$ is the resulting mapping of T_1 directly into T_3.

2.2 Colimits

The concept T_3 in Fig. 2 is more complex than the others, which is not surprising in view of the fact that the latter are the domains of morphisms for which T_3 is the codomain. This expresses the inheritance by T_3 of the information in the theories T_1 and T_2. Likewise, T_2 inherits from T_1. But there is more to be said about this hierarchical relationship among the theories, for our topic of interest concerns the manner in which a neural network learns, which we analyze as the formation of concept representations. The formation process involves the re-use of existing concept representations together with the new information supplied by the inputs. The two principal means for forming new concept representations are through abstraction—"pulling back" on a diagram of concepts and morphisms to extract a common concept—and specialization—"pushing out" on a diagram to

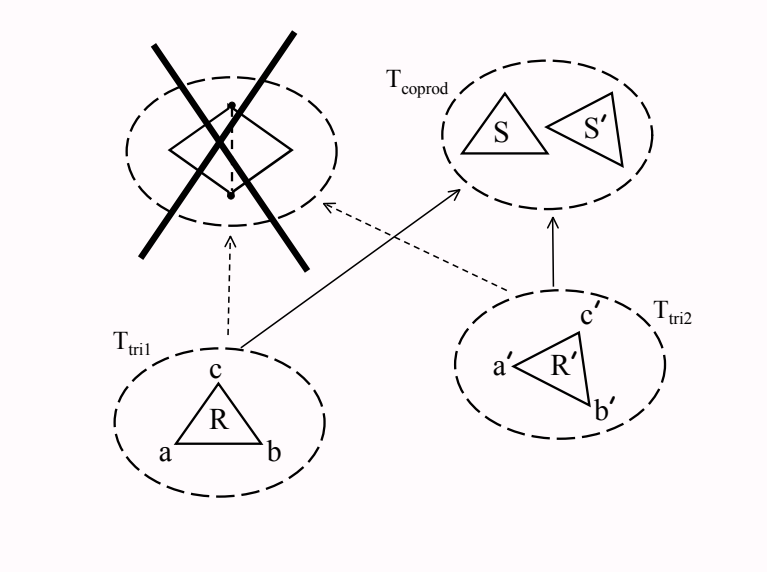


Figure 3: **Combining features in a set to obtain a composite object representation is ambiguous. It has no information concerning those parts of the two concepts T_{tri1} and T_{tri2} that are to be the same in the putative combination. What is actually represented is a coproduct, T_{coprod}.**

combine concepts in a structured manner using "concept blending". The categorical formalization of these two operations utilizes limits and colimits, respectively. In this paper, we focus upon colimits; as anticipated, they are here introduced with an example which is illustrated in two steps in Figs. 3 and 4. Briefly, a colimit is derived from a diagram of simpler concepts and morphisms, called its *base diagram*. In the derivation, a larger diagram is found (the *defining diagram* of the colimit) which contains the base diagram but which has two very special properties. Before illustrating this, it is worthwhile addressing the following question: Why not simply regard a concept specialization as a combination of concepts (its "features")? Why are the notions of "diagram", "morphisms in a diagram", and "colimit" necessary? To answer this question, let us examine an example of concept formation that relates more closely to the geometry of an object appearing in a visual field.

The intuition for Fig. 3 is entirely in terms of simple geometry. It illustrates the notion of *attempting* to represent a diamond shape by simply specifying that the shape consists of two triangles. Each triangle is expressed as a named object, R or R', together with its labelled vertices, sides, and so forth in its own copy of a theory expressing the geometry of triangles. The desired shape representation is envisioned as a combining of the two theories. But this specification lacks essential information, and this is revealed by formalizing the intuition using category theory. Unfortunately, the two theories together constitute a *discrete diagram*: a diagram with objects but no morphisms (except for the identity morphisms of the objects, which are always present and so are normally not shown). Because this diagram lacks information on how the objects within it are meant to be related in the combination, its colimit is a *coproduct*, consisting of two disjoint triangles as shown in the *valid* combination in Fig. 3 (the intended combination, which is not valid, is crossed out). In fact, formally, the coproduct theory contains two unrelated copies of the theories of points, lines, etc. which are necessary to express triangles, and, therefore, the combination involves two identical but unrelated notions of what constitutes a triangle. These facts are important, for consider the following. First, a mere combination of one or more components, or features, of an object in visual space is ambiguous, for the components can be combined in many ways to form a composite object. The same is true in any context; with no information for "blending together" entities, the notion of a combination of entities of any kind is meaningless. Second, the subject of our discussion is the structure of a neural network that has the capability of forming and recalling memories. The underlying assumption is that

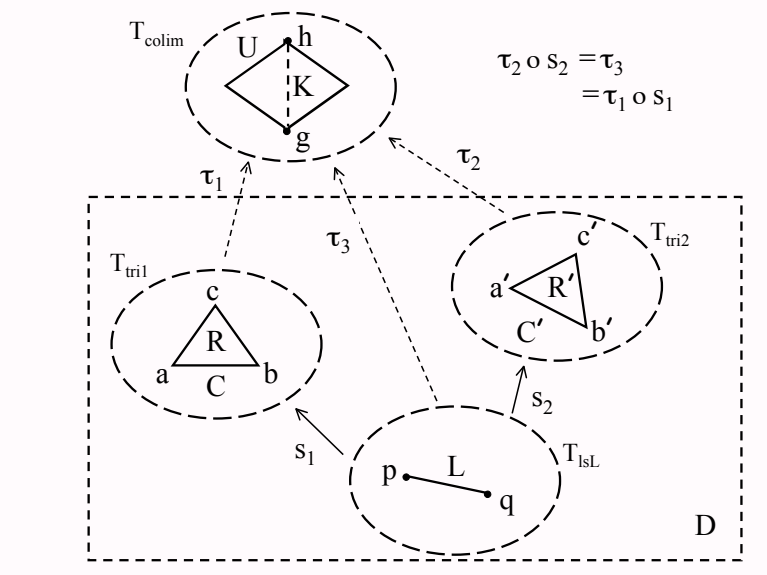


Figure 4: **A correct scheme for combining concepts. The combination has the needed information concerning those parts of the two concepts T_{tri1} and T_{tri2} that are to be the same in the putative combination T_{colim}.**

complex memories (of complex objects, of events, and of episodes consisting of sequences of events) consist of combinations or associations of simpler items. But if the neural network is to build these complex memories, it must somehow obtain the knowledge allowing it to form the specific combinations to correctly represent the objects, events, and episodes. Whether artificial or biological, a neural network is a sort of machine and cannot be assumed simply to have the required knowledge for any desired representation. It must have a mechanism for the expression of the requisite knowledge for "blending" or "pasting" components to form a composite.

The difficulties posed by the foregoing manner of representation of complex objects in memory is entirely due to the notion of memories formed as combinations over a collection of objects or features with no specific expressed relationships. Category theory offers an immediate alternative, one that expresses combinations along with information about the way in which the combined items are related in a particular combination—how they are "pasted" or "blended together". This alternative is the colimit construction.

In Fig. 4, a *cocone* is shown consisting of the morphisms τ_1, τ_2 and τ_3 along with their shared codomain T_{colim}, attached to the diagram D (the base diagram) having objects $T_{\text{tri1}}, T_{\text{tri2}}$ and T_{lsL} and morphisms $s_1 : T_{\text{lsL}} \longrightarrow T_{\text{tri1}}$ and $s_2 : T_{\text{lsL}} \longrightarrow T_{\text{tri2}}$. Here, the line segment L in the theory T_{lsL} is mapped via s_1 to the side C of the triangle R and via s_2 to the side C' of the triangle R' (this information can be written in a formal language related to the formal logic of the theories). The cocone has the property that all "triangular-shaped" diagrams involving a composition $\tau_i \circ s_i \, (i \in \{1, 2\})$ and another cocone "leg" τ_k with the same domain and codomain as the composition, commute. The domain and codomain in question are T_{lsL} and T_{colim}, and τ_3 is the "third leg" of two triangular diagrams; the other two legs of each are s_1, τ_1 and s_2, τ_2, respectively. Since each of these diagrams must commute, we have $\tau_1 \circ s_1 = \tau_3$ and $\tau_2 \circ s_2 = \tau_3$, which implies $\tau_1 \circ s_1 = \tau_2 \circ s_2$. A colimit is given by an *initial cocone*, an initial object in the category of cocones over D (for a definition of this category, see [10]). For our purposes, the significance of these two properties is as follows. The commutative property implies that regardless of the fact that the line segment L in the theory T_{lsL} is mapped into two separate geometric constructs via s_1 and s_2, it can be mapped to only one item in the colimit theory via the two compositions $\tau_1 \circ s_1$ and $\tau_2 \circ s_2$, for they are one and the same morphism. This implies that in the colimit object T_{colim} the two triangles

R and R' are "blended together" along the image K of the line segment L. Initiality implies that the colimit cocone must be the domain of a unique morphism for any other cocone for D having the latter as codomain. This means that the colimit cocone must be minimal, with its object having the least amount of information over all cocones for D; in effect, all other cocones "contain" it. To put this another way, the colimit object T_{colim} is a sort of least upper bound for the concepts that can be formed and include the concepts in D blended as indicated by s_1 and s_2. A neural network that can adapt its weights to form colimits for perceptual situations represented by diagrams has a powerful capability for concept representation, for it can form "canonical" concepts to represent the situations through concept blending, re-using concepts and relationships already available.

As just shown, two commutative triangles τ_1, s_1, τ_3 and τ_2, s_1, τ_3 sharing a common side can be "pasted together" to form a commutative square. When τ_1, τ_2, τ_3 form an initial cocone, the commutative square is the defining diagram for a *pushout*. Conversely, a colimit for a finite diagram can be decomposed into successive pushouts, each combining pushout (colimit) objects from the pushouts in the previous step. Hence forth, when discussing a colimit we shall often refer to the commutative triangles of its defining diagram, or to the commutative squares they form. We shall frequently refer to concepts such as T_{lsL} in Fig. 4 as "blending objects" in either the base diagram (such as D) or the defining diagram of a colimit, and to the colimit object as a "blending of concepts along shared subconcepts", keeping in mind that concept morphisms are not restricted to part-to-whole (subconcept) relationships.

This leads to the notion of concept representation in a neural network. As previously mentioned, this is expressed mathematically as a functor from the concept category to a neural category. Before discussing this in detail, an explanation of the derivation of a neural category is in order.

2.3 Neural categories, functors, and knowledge representation

In the CNST, in a well-designed neural network adaptation in response to stimuli results in the derivation of new concept and concept morphism representations through the re-use of existing representations. Before any adaptation has taken place, "pre-wired" nodes and connections at and near the sensor level of processing confer upon the network the ability to represent primitive or "perceptual" concepts. These describe the basic stimuli associated with sensor elements and some structures expressing specific combinations of sensor elements. The stimuli activate further neural structures, and the connection weight adaptation that follows forms new representations at a more complex level. As described in the colimit concept combination of the preceding section, the new representations extend the diagrams associated with sensor-level combinations by "pushing out", forming cocone representations by recruiting new neural network nodes through the formation of patterns of strengthened and weakened connections. Further extensions occur at more complex, but also at simpler, levels by both "pushing out" (concept specialization) and "pulling back" (concept abstraction). To formalize this process in correspondence with the concept objects and morphisms, we require a category that expresses limits (abstractions) and colimits (specializations), and this depends upon the neural network. If a category with sufficient structure can be derived for a given neural network, that network can then be shown mathematically to be capable of learning concept representations incrementally starting with a set of basic, sensor-level (perceptual) concepts and morphisms. The learned concepts will either inherit (as colimits) or abstract (as limits) information from the concepts and morphisms in the original diagrams from which they were formed. During retrieval, the appropriate limit and colimit representations will be re-activated depending upon the relative amounts of activation in the neural representation of their defining diagrams. This process forms a knowledge representation incrementally. We can formalize the achieved representation at each stage of adaptation in terms of a functor.

Specifically, during connectionist learning in a neural network A, connection weight adaptation results in a weight array w. In a properly designed neural network, this will result in one or more commutative diagrams associated with limits and/or colimits in the neural category $\mathbf{N}_{A,w}$ which is derived from A and w. We analyze the representation of concepts and their morphisms, diagrams, limits, and colimits at any stage w of adaptation in the network by attempting to define a functor $M\colon \mathbf{Concept} \longrightarrow \mathbf{N}_{A,w}$. If the neural network A with weight array w has a structure with the requisite properties, so that $\mathbf{N}_{A,w}$ has objects, morphisms, diagrams, and limits and colimits to match those of interest in **Concept**, then the functor M can be identified—at least the part corresponding to

the concept structures of interest. Since functors are many-to-one mappings, mapping all of **Concept** to $\mathbf{N}_{A,w}$ may be possible by "compressing" the unrepresented concepts and morphisms onto "compression objects" and morphisms in $\mathbf{N}_{A,w}$. Where the neural structures for representing concepts and morphisms of interest are missing from the architecture, a knowledge representation deficit has been found. That is, the network, at least with the current weight array, is incapable of representing that part of the structure in **Concept** and, hence, is not capable of responding to items in the input environment that are described by the missing concepts and morphisms.

One use of the CNST is in analyzing a neural architecture A, as above, to determine whether certain kinds of representations are even possible and, if so, where in the network they can occur, either through adaptation or "hard-wiring". Some "hard-wired" or preset structure serves as a basis for initiating the process of limit and colimit derivations by providing ready-made morphisms for diagrams at the input (and perhaps also at the output) interface. In fact, as shown in [7], these preset structures, when added to a neural network that does not have them, can yield improved input representations and thereby yield improved performance. We propose that in a biological neural system these preset structures arise during the critical period of brain development and make possible the perception of sensor primitives, for example, color and brightness in the primate visual system.

The CNST also can be applied to the design of new neural networks, suggesting the inclusion of neural structure which encompasses, for example, colimits and limits. This brings us to the subject at hand.

2.4 Neural categories

A neural network architecture A is given in the usual fashion by nodes p_i $(i = 1, 2, \ldots, n_n)$ and weighted connections c_j $(j = 1, 2, \ldots, n_c)$ which at a given stage of adaptation have weight values w_j. At any stage of adaptation, with weight array w, the neural network has an associated category $\mathbf{N}_{A,w}$. The category is a mathematical representation of the structure comprising the network connections, its current weights, and its potential activity patterns given the weight array w. The nodes and connection pathways are called *carriers* of the objects and morphisms of $\mathbf{N}_{A,w}$. A morphism is defined in terms of a set of parallel connection paths that share the same source and target nodes, where the source node is the carrier of its domain and the target node is the carrier of its codomain. The morphism has a set of *instances*, states of activity over the network in which the outputs of the nodes in the path set of the morphism vary *simultaneously* within some tolerance of their initial values. This is assumed to occur during a time interval sufficiently long to be significant (for example, connection weight adaptation can occur to exploit the information content of this activity). More precisely, the instances of a morphism are the elements of the intersection of the instances of its objects, which in turn are the sets of activation states of their carriers.

To establish notation, an object of $\mathbf{N}_{A,w}$ is a pair (p_i, η); its carrier p_i is a node of A and η is one of possibly many sets of signal function output values for p_i. If the signal function is ϕ_i and θ is an activation state yielding an output in η for p_i, we can write $\phi_i(\theta_i) \in \eta$, where θ_i is the component of θ associated with p_i. The assortment of intervals η associated with a given node and the type of their values are options for the CNST analyst, within reasonable limits. For example, η must have a structure—usually, that of a number system—that allows elements to be regarded as "close to" each other and that admits certain algebraic operations, such as addition for the purpose of accumulating input sums at a node. Often, signal function outputs and connection weights will be considered as real values, and the sets η will then be intervals of real values. An alternative type for values is the complex number system, in which case each η is a region in the complex plane. Real quantities will be assumed here.

Because we are interested in situations involving connection paths whose nodes are experiencing simultaneous activations with minimal variation, we assume that the nodes of current interest have activation values varying within some tolerance around their initial values θ_i over the current time interval. In particular, when the object (p_i, η) is participating in an instance of a morphism (to be defined), this means that $\phi_i(\theta_i) \in \eta$. It is important to note that nodes not in the subnetwork of current interest can have their activation values, hence their outputs, vary significantly as the network undergoes weight adaptation. The initial conditions for a time interval associated with an instance of (p_i, η) are denoted (θ, e), where e is a neural network input pattern occurring along with the

state θ. We write $(\theta, e) \in U_{(p_i,\eta),w}$, where $U_{(p_i,\eta),w}$ is the instance set for (p_i, η) when the weight array is w.

A *connection path* is a chain of connections along with their source and target nodes; we can express this in list notation, as $[p_1, c_1, p_2, c_2, p_3, c_3, p_4, \ldots p_k]$ (the selection of indices $1, 2, 3, \ldots$ here is for simplicity). Because we are interested in the activity as well as the connection structure of a network, for the same reason that neural objects depend upon node outputs as well as the nodes themselves, a neural morphism must take account of the outputs for the nodes lying in a connection path. Thus, we use *signal paths*, which specify sets of outputs for the nodes. A signal path based upon the aforementioned connection path (again, in list notation and using an ordered set of indices for simplicity) appears as $[(p_1, \eta_1), c_1, (p_2, \eta_2), c_2, \ldots (p_k, \eta_k)]$, where the η_i are the specified output sets for the path nodes. This signal path can be used to define a morphism with domain object (p_1, η_1) and codomain object (p_k, η_k). It is important to discuss instances again before pursuing this thought.

Recall that an instance of any of the items discussed here is an occurrence over the entire network that produces an output or outputs in the specified output set or sets of the item(s). Applying the principle of simultaneity, an instance (θ, e) for a signal path μ, with $(\theta, e) \in U_{\mu,w}$, is simultaneously an instance of all objects in the path. Therefore, $U_{\mu,w} = U_{(p_1,\eta_1),w} \cap U_{(p_2,\eta_2),w} \cap \ldots U_{(p_k,\eta_k),w}$. Now, the fact that μ incorporates neural objects other than the domain and codomain of its morphism suggests that there can be many morphisms each associated with a different one of its segments. In fact, there is a morphism associated with, for example, the single-connection path $[(p_1, \eta_1), c_1, (p_2, \eta_2)]$ which is a segment of μ. Its instance set is $U_{(p_1,\eta_1),w} \cap U_{(p_2,\eta_2),w}$, since every instance in which p_1 and p_2 are generating outputs within the indicated intervals η_1 and η_2 is an instance of this path. Not only are there morphisms associated with the segments of μ, but also with signal path sets containing μ in which all members have the objects (p_1, η_1) and (p_k, η_k) as domain and codomain. Given a set Γ of such signal paths, its instance set is determined, again by simultaneity, as the intersection of the instance sets for its members. In Fig. 5, two signal paths γ and γ' are shown connecting objects (p_1, η_1) and (p_4, η_4), with $\gamma = [(p_1, \eta_1), c_1, (p_2, \eta_2), c_3, (p_4, \eta_4)]$ and $\gamma' = [(p_1, \eta_1), c_2, (p_3, \eta_3), c_4, (p_4, \eta_4)]$. In the figure, the partially-colored-in vertical bars represent the level values of the current outputs within the intervals η_i at the nodes p_i. Either path γ or γ' defines a morphism by considering either $U_{\gamma,w}$ or $U_{\gamma',w}$ only . On the other hand, the path set Γ, where $\Gamma = \{\gamma, \gamma'\}$, is associated with a morphism we shall designate as m_5, as shown in the figure. The instance set $U_{\Gamma,w}$ of the morphism uniquely associated with Γ is $U_{\Gamma,w} = U_{\gamma,w} \cap U_{\gamma',w}$.

The two paths in Fig. 5 from (p_1, η_1) to (p_4, η_4) each consists of two connections through a third object. In fact, this suggests our definition of composition of morphisms for $\mathbf{N}_{A,w}$. The single-connection path γ_1, where $\gamma_1 = [(p_1, \eta_1), c_1, (p_2, \eta_2)]$, is uniquely associated with a morphism m_1, and γ_3, where $\gamma_3 = [(p_2, \eta_2), c_3, (p_4, \eta_4)]$ is associated with a morphism m_3. Concatenating the two yields the two-connection path γ, uniquely defining a morphism m. The instance set of γ, and, hence, of m, is $U_{\gamma,w} = U_{\gamma_1,w} \cap U_{\gamma_3,w}$. We define composition in $\mathbf{N}_{A,w}$ so that $m = m_3 \circ m_1$.

Similarly, the other two-connection path γ' uniquely defines a morphism $m' = m_4 \circ m_2$ with instance set $U_{\gamma',w} = U_{\gamma_2,w} \cap U_{\gamma_4,w}$. In general, m and m' can be two separate morphisms, and as before Γ has the instance set $U_{\Gamma,w} = U_{\gamma,w} \cap U_{\gamma',w}$, associated with the morphism m_5. *However, if $U_{\gamma,w} = U_{\gamma',w}$, then the diagram formed by $m_1, m_2, m_3, m_4, \underline{m}$ is commutative*, $m_3 \circ m_1 = m_5 = m_4 \circ m_2$, because then $U_{\gamma,w} = U_{\Gamma,w} = U_{\gamma',w}$.

It is important to emphasize that a node can be the carrier of several neural objects. That is, any of the nodes p_i in Fig. 5 can have several objects of the form (p_i, η_i) associated with it; in a real-valued neural network model, η_i can be any of several, possibly infinitely many, intervals. Purely to simplify the notation for objects in the present discussion, the intervals have been given subscripts identical with those for the nodes, as for example (p_4, η_4). Further simplifications will appear in the following discussion. For example, properly speaking, it is neural network nodes p_i that are the carriers of objects (p_i, η_i), and not the objects themselves, that become active (or activated), which causes the nodes to generate outputs ξ in the intervals $(\xi \in \eta_i)$. To avoid lengthy definitions and more terminology, however, we shall say also that objects, connections, and signal paths, morphisms, and diagrams are active when the appropriate nodes are active.

Another simplification has been used in Fig. 5 and will occur throughout, to wit: The only nodes and connections shown in a figure are those necessary to illustrate the point being made. In actuality, a node can have inputs and outputs that are not shown. The latter may be necessary to help bring about the activations that we call

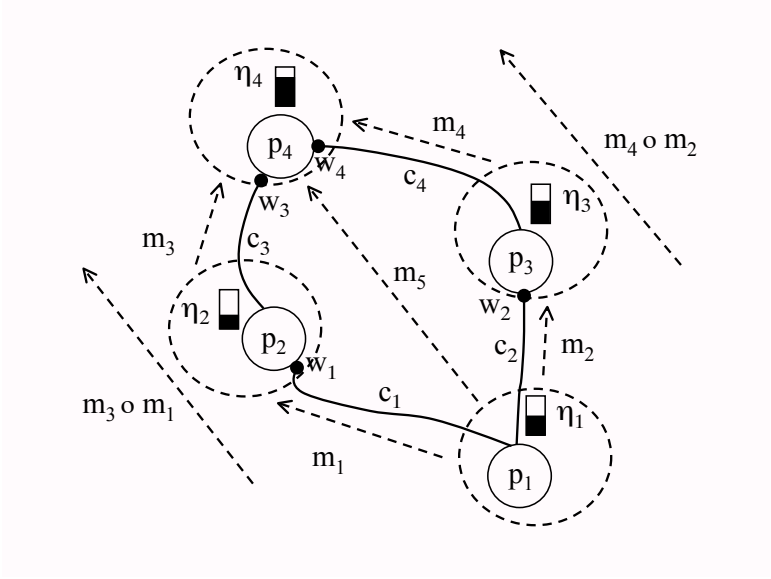


Figure 5: **There are four neural morphisms m_1, m_2, m_3, m_4 defined by the single-connection paths for connections c_1, c_2, c_3, c_4 between the neural objects $(p_1, \eta_1), (p_2, \eta_2), (p_3, \eta_3), (p_4, \eta_4)$, with compositions $m_3 \circ m_1$ and $m_4 \circ m_2$ both having the same domain (p_1, η_1) and codomain (p_4, η_4). A third morphism m_5 with the same domain and codomain is defined by the set containing both of these paths. The diagram commutes if $m_3 \circ m_1 = m_5 = m_4 \circ m_2$.**

instances of the neural objects and morphisms; for example, activity in node p_1 certainly requires an excitatory input assuming that it is not tonically active.

2.5 Functors

Mathematically, we analyze concept representation in a neural network A at a given stage of weight adaptation w as a functor M: **Concept** $\longrightarrow \mathbf{N}_{A,w}$. Figure 6 shows a functorial mapping of concepts T_1, T_2 and T_3 and a composition $s_2 \circ s_1 : T_1 \longrightarrow T_3$ of concept morphisms $s_1 : T_1 \longrightarrow T_2$ and $s_2 : T_2 \longrightarrow T_3$ to neural objects (p_1, η_1), (p_2, η_2) and (p_3, η_3) and a composition $m_2 \circ m_1 : (p_1, \eta_1) \longrightarrow (p_3, \eta_3)$ of neural morphisms $m_1 : (p_1, \eta_1) \longrightarrow (p_2, \eta_2)$ and $m_2 : (p_2, \eta_2) \longrightarrow (p_3, \eta_3)$. The domain T_1 of s_1 is a theory of triangles; the codomain T_2 is a theory of a specific isosceles triangle given the name R. In terms of the pictorial illustrations with which theories and morphisms are to be illustrated, s_1 can be thought of as an explanation of *how* the theory of triangles is used in describing an isosceles triangle—why the latter appears as it does (that is, what constitutes a triangle). Similarly, the codomain T_3 of s_2 is a theory of a weight with a triangular cross-section resting on a horizontal surface; s_2 describes the incorporation of the theory of R into its codomain, where the image of R expresses the triangular cross-section of the weight.

For simplicity, m_1 and m_2 are associated with single-connection signal paths, $[(p_1, \eta_1), c_1, (p_2, \eta_2)]$ and $[(p_2, \eta_2), c_2, (p_3, \eta_3)]$, respectively. The functorial property specifies that the image of a composition is the composition of images. In the example of Fig. 6, written in shorthand (without the domains, codomains, and arrows), this is expressed $M(s_2 \circ s_1) = M(s_2) \circ M(s_1) = m_2 \circ m_1$. The functorial property ensures that the compositionality of relationships between concepts is preserved in their neural representations. Their composition $m_2 \circ m_1$ is shown associated with two signal paths: the composite path $[(p_1, \eta_1), c_1, (p_2, \eta_2), c_2, (p_3, \eta_3)]$ and an additional path $[(p_1, \eta_1), c_3, (p_3, \eta_3)]$. This illustrates the fact that the set of instances of a concatenation of

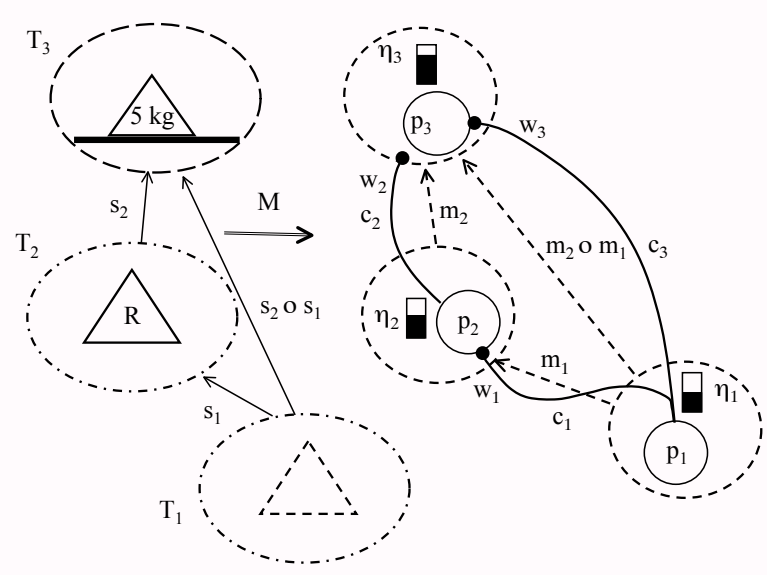


Figure 6: **Functorial mapping of a composition of concept morphisms to a composition of neural morphisms.**

paths may be shared by other paths having the same source and target objects. It also illustrates a principle we propose: that in many important situations involving the composition of neural morphisms representing concept morphisms, the path set of the composite includes an additional path that increases the likelihood that an instance of the domain of a composition will correspond with an instance of the codomain. Furthermore, the concept morphism representation includes a reciprocal path with connection strength sufficient to ensure that the Model-space Morphism Principle [10] is followed: An instance of the codomain entails an instance of the domain. For simplicity, the reciprocal path is not shown.

3 Temporal Sequences

Fig. 7 illustrates a temporal sequence expressed as a colimit for a diagram in the category **Concept**. Here, the event sequence is described piecemeal by two concepts, $T_{\texttt{event1}}$ and $T_{\texttt{event2}}$, expressing the temporal sequence of the triangular weight falling a short distance onto the horizontal surface. The colimit concept $T_{\texttt{episode}}$ expresses the sequence in full; to be a correct representation, it must include the information that the shape of the weight in both $T_{\texttt{event1}}$ and $T_{\texttt{event2}}$ is described by a single concept $T_{\texttt{tri-wt}}$. The diagram of four morphisms in the figure is a pushout square and, hence, commutes, indicating that it involves only a single morphism with domain $T_{\texttt{tri-wt}}$ and codomain $T_{\texttt{episode}}$. This expresses the correct blending of $T_{\texttt{event1}}$ and $T_{\texttt{event2}}$ along their shared subconcept $T_{\texttt{tri-wt}}$. Clearly, the concepts in the diagram inherit information from concepts and morphisms other than those shown in the diagram, such as time, distance, mass, gravity, and velocity. Nevertheless, Fig. 7 is sufficient to illustrate the point about combining concepts properly as was discussed in the example of Figs. 3 and 4. The episode representation does, however, introduce the notion of time into our analysis, and this requires a special treatment.

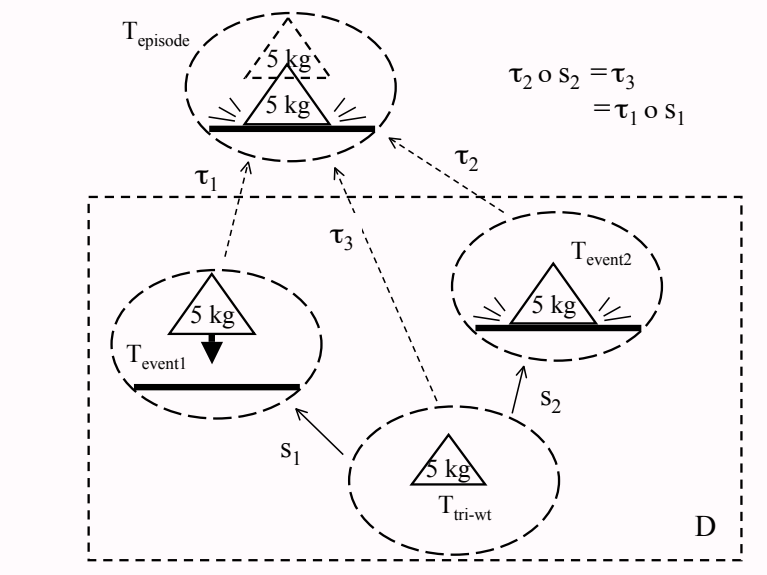


Figure 7: **A colimit representing an episode (a falling weight hitting a solid surface), expressing the episode unambiguously as a blending of two events along a shared subconcept (the triangle concept).**

3.1 Time-Dependent Representations of Temporal Stimuli

In previous discussions involving the CNST, the neural network concept representations have been time-independent. To express the representations of temporal sequences of events in the theory, the definition of neural morphism must be extended. This is because of the exclusive dependence of the existing definition on the notion of simultaneity. Of course, an event occurring over time can be represented in the existing theory as seen in Fig. 7, but only in a compressed fashion. Since the major concepts expressing different time steps of a memory (events 1 and 2 in Fig. 7) must be included in the diagram from which the colimit is derived, and since a commutative diagram in a neural category is represented by the simultaneous activation of neural network nodes along signal paths that are the carriers of the morphisms of a diagram, everything involved in the event is represented as a single occurrence, a "snapshot". If the event is of short enough duration this might not be a serious drawback. The stored memory represented by the colimit will be analogous to either an event recorded on slow film or in a digital medium with a slow frame rate, or, alternatively, as a lossy compression whereby only certain parts of the event occurring at different times are represented. However, this scheme does not allow the CNST to express temporal sequences of such duration that it is important that they be replayed over time. It does not allow the retrieval of the *neural representation* of the colimit concept to *play back the temporal sequence*.

To address this issue, we have extended the notion of neural morphism by generalizing the notion of simultaneity. In terms of Fig. 7, the issue to be resolved is that the two concept morphism compositions $\tau_1 \circ s_1$ and $\tau_2 \circ s_2$ must be functorially mapped to neural morphism compositions whose signal paths can be active at different times instead of simultaneously. Yet, to be functorial, the mapping from concept to neural category must map the commutative concept diagram to a commutative neural category diagram, and this involves the notion of simultaneity.

Were the two compositions in Fig. 7 mapped to the two compositions $m_3 \circ m_1$ and $m_4 \circ m_2$ in the commutative neural category diagram of Fig. 5 (a convenient example since this diagram has the same trapezoidal shape), what would be required is that $m_3 \circ m_1 = m_4 \circ m_2$ regardless of the fact that the two compositions were allowed to be activated separately. Now, the foregoing equation states that there is a single morphism $\underline{m}$ associated

with this diagram that has domain (p_1, η_1) and codomain (p_4, η_4). But there could be other neural morphisms with the same domain and codomain, representing other concept morphisms with domain $T_{\texttt{tri-wt}}$ and codomain $T_{\texttt{episode}}$ but not involved in this diagram. If we loosen the notion of simultaneity by allowing the two paths to be active at different times, how are we to distinguish morphisms with more than one path from separate morphisms associated with separate paths which happen to share a common domain and codomain? Simply allowing the separate activation of the signal paths involved in a morphism without some restriction loses the information necessary to identify the separate activations as a single instance, that is, an instance of a single morphism.

3.2 Temporal Neural Morphisms

The solution to this problem requires an unambiguous means of redefining simultaneity that allows separate signal paths of a morphism to be active at different times, yet ensures that the separate activations together constitute an instance of a single morphism. We proceed as follows. First, we require that the neural representations of the domain and codomain objects of a morphism remain active throughout an instance; this expresses continuity in time. The intermediate nodes along the separate signal paths can be allowed to have separate activations, since the continuous activity associated with the domain and codomain maintains the continuity of the instance (an additional means of maintaining continuity will be added presently). Also—and this is the second part of the solution—the activations of the objects (p_i, η_i) within each signal path must as before occur simultaneously, because at the most elementary level this distinguishes a signal path; notice that this requirement is consistent with the first part, for the domain and codomain objects must remain active. Finally, the paths must become active in a continuous sequence, one after the other, during the instance.

These requirements impose a rather challenging constraint on neural network design and adaptation, for they require a network to locally self-synchronize its activities along separate connection paths having a common source and target when the paths are separately active. We suggest a neural design principle to supply an architectural mechanism for synchronization: In a temporal morphism, an additional, persistently-active path is added to the set of separately-active paths of the morphism. The continuity of activity in the carrier nodes of the domain and codomain of a temporal neural morphism is thereby ensured by the presence of a continuously-active path connecting them. If this additional path is accompanied by a reciprocal connection, there can be feedback as well, enabling a mutually-supportive interplay of the activities of the domain and codomain carrier nodes. The feedforward path from domain to codomain is illustrated by the single connection c_3 in Fig. 6. Notice that the morphism associated with it was supposed to have the same instances as (hence, to be the same as) the composition morphism $m_2 \circ m_1$. In that case, we were proposing as a principle that such additional paths, formed perhaps through adaptation, accompany the concatenated paths of a composition of morphisms in certain important cases. A temporal neural morphism is such a case.

Figs. 8, 9 and 10 illustrate the replay of a temporal sequence represented by a neural morphism. The object (p_5, η_5) is a colimit object, the image of a concept colimit such as that in Fig. 7, for a diagram consisting of objects $(p_3, \eta_3), (p_4, \eta_4)$ and (p_0, η_0) and morphisms $M(s_1), M(s_2)$ associated with the signal paths γ_1, γ_2, respectively, where $\gamma_1 = [(p_0, \eta_0), c_1, (p_1, \eta_1), c_3, (p_3, \eta_3)]$, $\gamma_2 = [(p_0, \eta_0), c_2, (p_2, \eta_2), c_4, (p_4, \eta_4)]$ and M:**Concept** $\longrightarrow \mathbf{N}_{A,w}$ is the concept representation functor for the neural representation with the current weight array w for a neural network A. In addition to its apical object (p_5, η_5), the colimit cocone includes the leg morphisms $M(\tau_1), M(\tau_2)$ and $M(\tau_3)$ associated with c_5, c_6 and c_7 (actually, with the single-connection signal paths containing the connections). Just as with the concept category quantities they represent, the objects $(p_0, \eta_0), (p_3, \eta_3), (p_5, \eta_5)$ and morphisms $M(s_1), M(\tau_1), M(\tau_3)$ and $M(s_2), M(\tau_2), M(\tau_3)$ form a pushout square. Hence, $M(\tau_1) \circ M(s_1) = M(\tau_3) = M(\tau_2) \circ M(s_2)$, that is, all three morphisms with domain object (p_0, η_0) and codomain object (p_5, η_5) are one and the same, and therefore their associated signal paths are all part of the same morphism. However, the architecture has been arranged for a stepwise temporal replay: The composite path $\gamma_1; [(p_3, \eta_3), c_5, (p_5, \eta_5)]$ of the temporal morphism will become active first and the composite path $\gamma_2; [(p_4, \eta_4), c_6, (p_5, \eta_5)]$ will become active next, with the first path suppressed. The path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$ will remain active throughout, maintaining the continuity of the temporal morphism. The objects (p_1, η_1) and (p_2, η_2) are intermediate objects along the paths γ_1 and γ_2, respectively; an example of

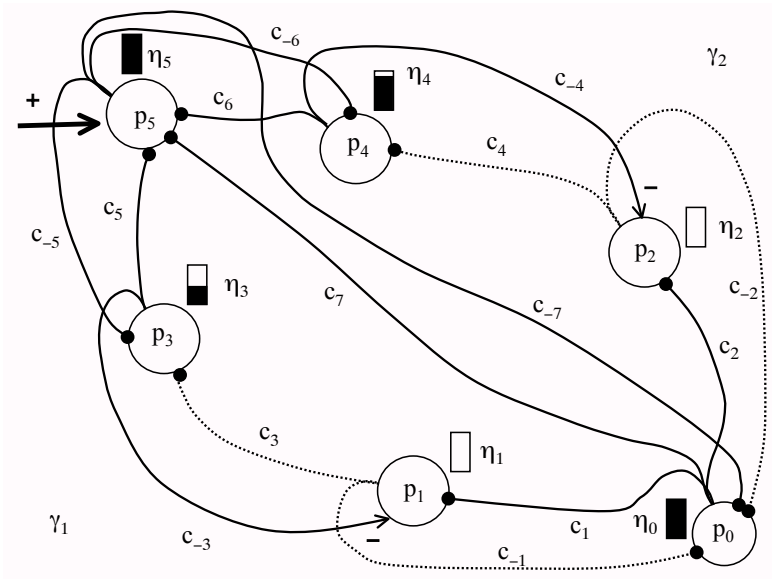


Figure 8: **A temporally extended neural morphism. Temporal replay during recall begins when an excitatory stimulus (+) activates the carrier of the colimit object** (p_5, η_5)**.**

such objects will be seen in the temporal sequence architecture to be presented in the next section. As in Fig. 5, the colored-in vertical bars represent the level values of the current outputs within the intervals η_i at the nodes p_i. For the present case, most of the intervals are regarded as containing all nonzero values. The exception is that the intervals η_3, η_4 are of the form $\eta_3 = \{\xi \mid 0 \leq \xi \leq t_1\}$ and $\eta_4 = \{\xi \mid 0 \leq \xi \leq t_2\}$, where the interval upper bound values t_1, t_2 are the heights of the bars for η_3, η_4 shown in Fig. 8. This allows these two intervals to represent the time t_i relative to the start of the temporal replay at which their nodes first are to become active. Notice that $t_1 \leq t_2$, indicating that node p_3 is to become active before node p_4. Because they signify in increasing order the relative times at which their nodes are to become active, the values t_1, t_2 form a *recency gradient*. Notice, finally, that each connection c_i has a reciprocal, denoted c_{-i}. All feedforward connections shown (oriented in the direction from p_0 to p_5) are excitatory, as are their reciprocals with the exception of reciprocals c_{-3} and c_{-4}.

The replay of the temporal sequence represented by the subnetwork in Fig. 8 begins with a stimulus that activates the node p_5, signified by the horizontal arrow labelled "+". This causes p_5 to reach its full excitation level; through the reciprocal connections c_{-5}, c_{-6} and c_{-7}, it stimulates the carriers p_0, p_3 and p_4 of the objects $(p_0, \eta_0), (p_3, \eta_3)$ and (p_4, η_4) in the base diagram of the colimit. Now, p_3 and p_4 are *temporal integrator nodes*; initially, they generate output magnitudes t_1, t_2 corresponding to the previously-mentioned recency gradient, where (p_3, η_3) is the object representing event 1 of the sequence and (p_4, η_4) represents event 2. This occurs because the same pattern of magnitudes was formed in the weights w_{-5}, w_{-6} of c_{-5}, c_{-6} when this two-step sequence was learned. The weight value w_{-7}, on the other hand, is unity, causing p_0 to become fully activated immediately following the activation of p_5. Now, because of the inhibitory feedback via c_{-3}, c_{-4}, the intermediate objects $(p_1, \eta_1), (p_2, \eta_2)$ of γ_1, γ_2 are suppressed following the activation of the integrator nodes and, hence, only the path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$ of the temporal morphism can be active. This is its initial state.

The replay of step one (event 1) of the sequence occurs during the next phase of activity, shown in Fig. 9. Because the temporal integration is a continuing process, the output of the nodes p_3, p_4 is continually decreasing; eventually, p_3 has reached a sufficiently low level that its output t acting through c_{-3} is no longer sufficent to suppress p_1, where now $t << t_1$. The object (p_1, η_1) now becomes active because of the continuing input from

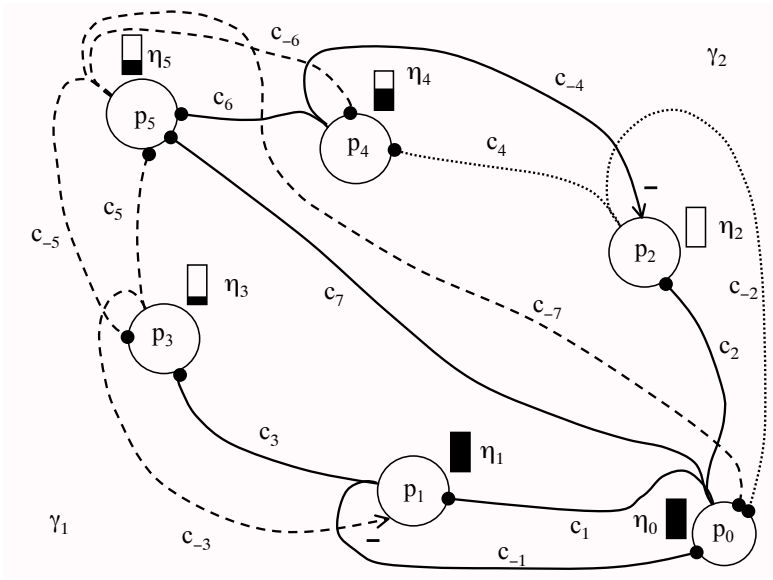


Figure 9: **A temporally extended neural morphism: First, one path bundle (consisting of a single path in this example) becomes active because the source node is firing and nothing is blocking the activation of the intermediate path nodes. However, the bundle on the right is temporarily blocked by inhibitory input to its intermediate nodes.**

the node p_0, allowing the path γ_1 to be active (note the solid line for c_3). Meanwhile, node p_5 has experienced an activity decay, making it less able to reinforce the activity of p_0, p_3, p_4 through its reciprocal connections to them. The dashed lines in Fig. 9 indicate the weak activity through those connections. Nevertheless, the activity through connection c_5 is strong enough to complete the first commutative triangle diagram, associated with the equation $M(\tau_1) \circ M(s_1) = M(\tau_3)$ and corresponding to activity in the paths $\gamma_1; [(p_3, \eta_3), c_5, (p_5, \eta_5)]$ and $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$. It is the dotted lines that indicate an inactive connection; for example, while both c_3 and c_4 were inactive during the intial state of the temporal morphism, only c_4 is inactive during step 1. Note the continued activity of the path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$.

In step two, shown in Fig. 10, activity in node p_4 has decayed to such an extent that node p_2 can become active. In similarity with step 1, in step 2 the activity through c_4 completes the second commutative triangle of the pushout square, associated with the equation $M(\tau_2) \circ M(s_2) = M(\tau_3)$ and corresponding to activity in the paths $\gamma_2; [(p_4, \eta_4), c_6, (p_5, \eta_5)]$ and $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$. The first commutative triangle is now inactive because the activity in node p_3 has decayed to a subthreshold level. Again, notice the continued activity of the path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$.

Because the diagram in Fig. 7 forms a pushout, the morphism τ_3 is equal to both of the compositions $\tau_1 \circ s_1$ and $\tau_2 \circ s_2$. Mathematically, it is each morphism, and all three morphism symbols are simply different ways of denoting the same thing. The same is true of the morphism $M(\tau_3)$: it is simply the two compositions $M(\tau_1) \circ M(s_1)$ and $M(\tau_2) \circ M(s_2)$, for again these are three ways of denoting the same thing. In principle, then, the connection c_7 in this temporal morphism example is unnecessary, for the composite paths $\gamma_1; [(p_3, \eta_3), c_5, (p_5, \eta_5)]$ and $\gamma_2; [(p_4, \eta_4), c_6, (p_5, \eta_5)]$ are carriers of the morphism $M(\tau_3)$, by definition. Also in principle, however, we require a full formalization of our definition of temporal morphism, one that not only supports temporal replay but also distinguishes a temporal morphism from separate morphisms with the same domain and codomain but with path sets whose instances occur at separate times. In the example just given, including the continuously-active path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$ serves both purposes: It maintains the continuity expressing an extended notion of an instance of a morphism while at the same time providing a means of identifying a temporal morphism in con-

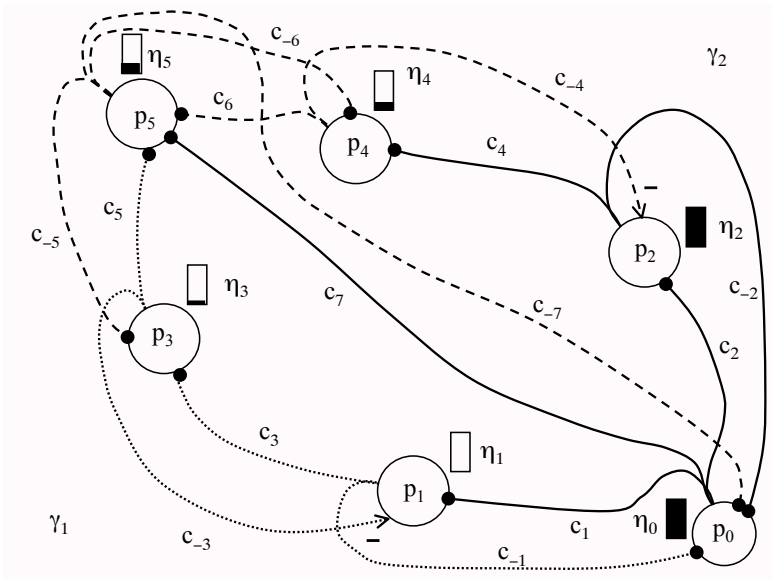


Figure 10: **As the memory sequence progresses, the other path bundle becomes active while the first bundle is blocked. The source and target nodes remain active during both phases of the temporal sequence.**

tradistinction to non-temporal structures which are otherwise very similar. In fact, this additional path is identified as a colimit leg morphism, τ_3 (or $M(\tau_3)$) in this case. This suggests extending our new design principle to all colimits, as follows: Each colimit leg morphism includes in its path-set its own unique path. As will be seen, the inclusion of this path is helpful not only in formalization and in temporal replay, but in the adaptation through which a temporal sequence representation is first formed; more generally, it can aid in the formation of colimits.

3.3 Learning an Episode: Incremental Colimit Formation

Connection-weight adaptation in the subnetwork of Figs. 8–10 in response to the event sequence enables the later replay of the sequence as described. The adaptation results in the formation of a temporal colimit with apical object (p_5, η_5). The temporal morphism just described is the composition along the sides of a pushout square in the defining diagram of this colimit. The question of how this adaptation can occur will be addressed in Section 4, where we describe an experimental architecture.

Fig. 11 shows the same episode concept as Fig. 7, but formed incrementally by first deriving $T_{\texttt{event1}}$ and $T_{\texttt{event2}}$ as colimits of diagrams D_1 and D_2 in which two concepts of shapeless items with mass M are merged with the concept of a particular triangle to form the two separate events. Diagram D expresses the derivation of $T_{\texttt{episode}}$ by merging the two events along their common triangular shape concept. Fig. 12 again shows the colimit derivations for the two diagrams D_1 and D_2 in Fig. 11, but without pictures. Expressing diagrams in this purely notational format will simplify the expression of diagrams having greater complexity, including, for example, concepts and morphisms relating to time. An elementary concept of time would be the common domain of morphisms whose codomains are concepts which express specific times. Adding these yields the diagrams illustrated in Fig. 13. Even though we have designed an architecture with a temporal integrator, as mentioned in the Introduction, we have not yet formulated an explicit representation of knowledge about time (formulating useful concepts about time raises some fundamental issues). For this reason, the diagrams in Fig. 13 are vague about this knowledge, indicated by the legend "theories of time". The figure is included merely to indicate what is needed for a full theoretical treatment of temporal sequences in the CNST. In the architecture to be described,

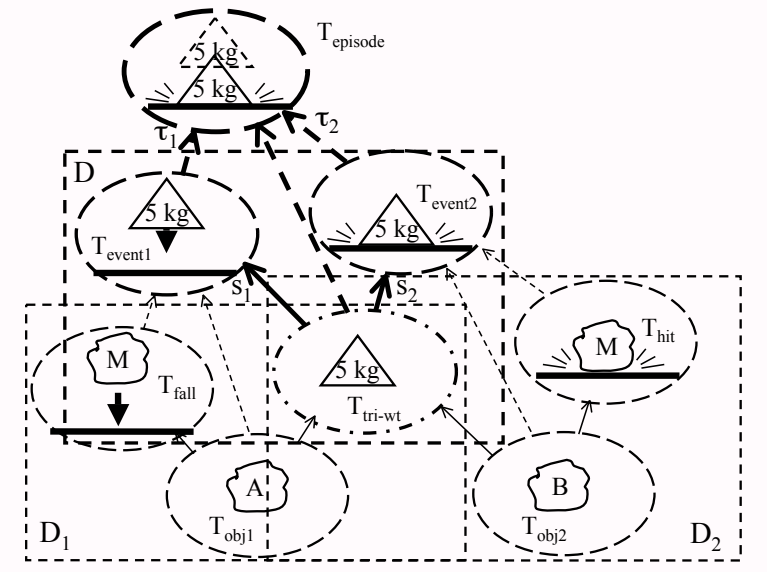


Figure 11: **A hierarchy of colimits builds an episode theory.**

the temporal integrator is assumed simply to subsume the notion of time through its behavior, which involves the notion of temporal morphisms discussed in the preceding section. Notice also the mention of coproducts. These are colimits of diagrams having no morphisms—hence, no blending. This is acceptable for combining concepts which are completely unrelated, but note that time as well as the other theories in the diagrams do have subconcepts in common: those relating to the theory of numbers. A full treatment would include in the diagram a concept of number as a blending object (number as represented by the neural network, not necessarily a full theory of numbers) and appropriate morphisms having it as domain. Finally, the episode theory is formed in a two-step process as indicated in Figs. 14 and 15.

Although a concept of time is not explicit in the present formulation, notice that its inclusion is suggested by the temporal integration in Figs. 8–10. At the time the sequence is learned as a colimit, the integrator nodes have relative output magnitudes that form a recency gradient. The concept of "time t_1"—the time at the occurrence of the first step, relative to any time at which the sequence is started— is represented in the object (p_3, η_3); if the magnitude of output indicated by the bar in Fig. 8 has the value t_1, then the interval η_3 is $\eta_3 = \{\xi \mid 0 \leq \xi \leq t_1\}$, as in the preceding section. In Fig. 13, this concept is labelled T_{t_1}. Having diagrams available for colimit constructions such as these allows us to see how such concept representations can be formed in the neural network, and what consequences this has for neural structure.

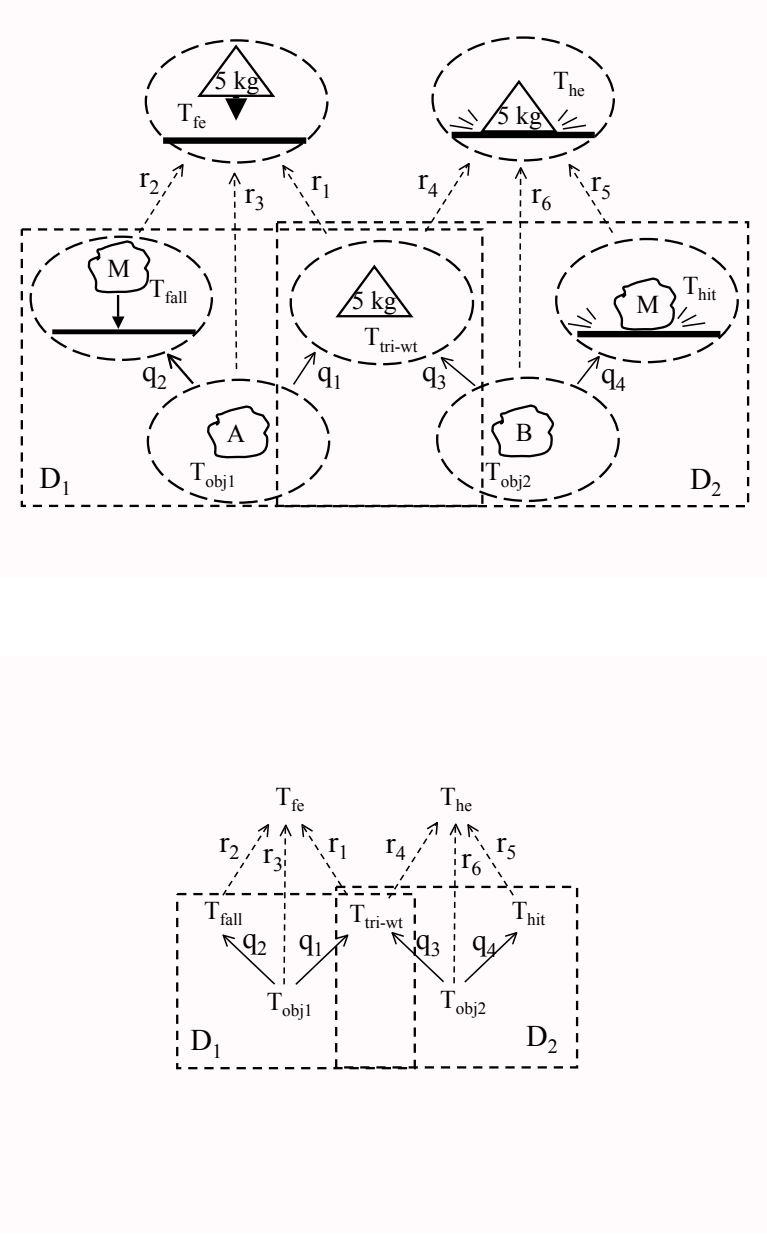


Figure 12: **Colimits of two diagrams form the objects for the two events. The theories are illustrated both pictorially (top) and by their labels (bottom).**

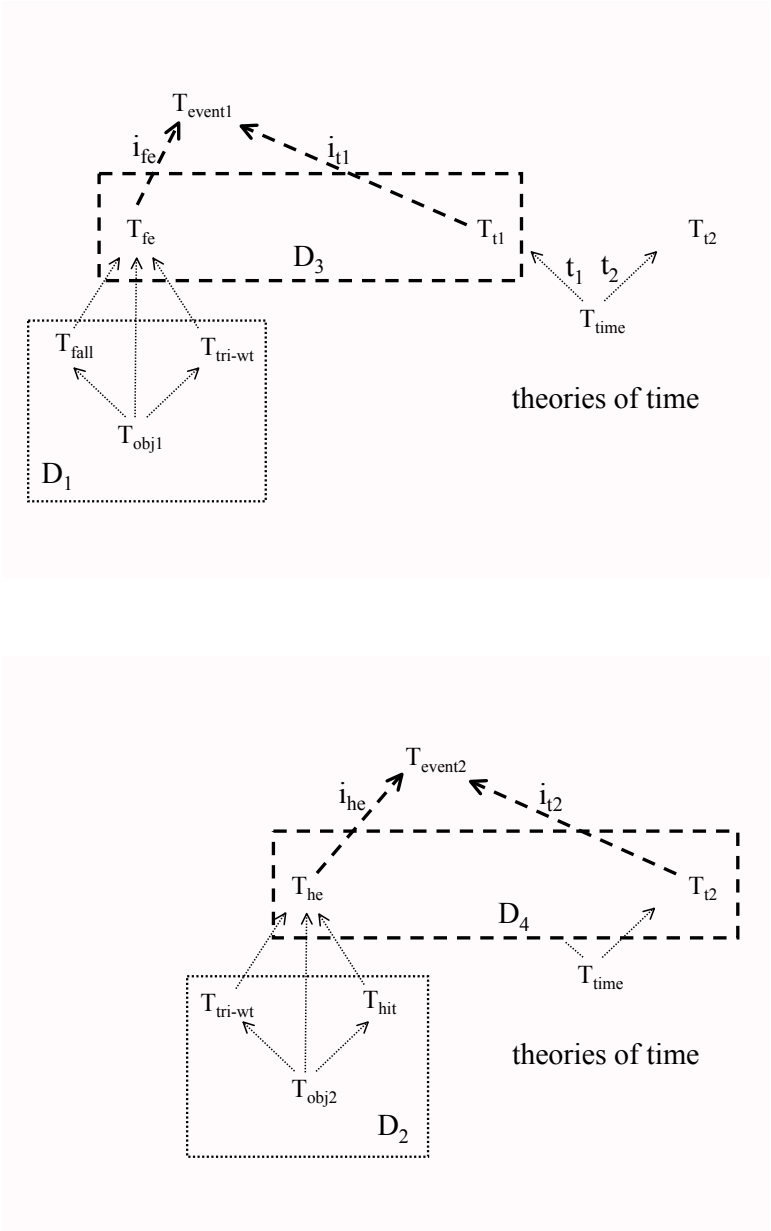


Figure 13: **The two events are shown as coproducts that combine items with time of occurence.**

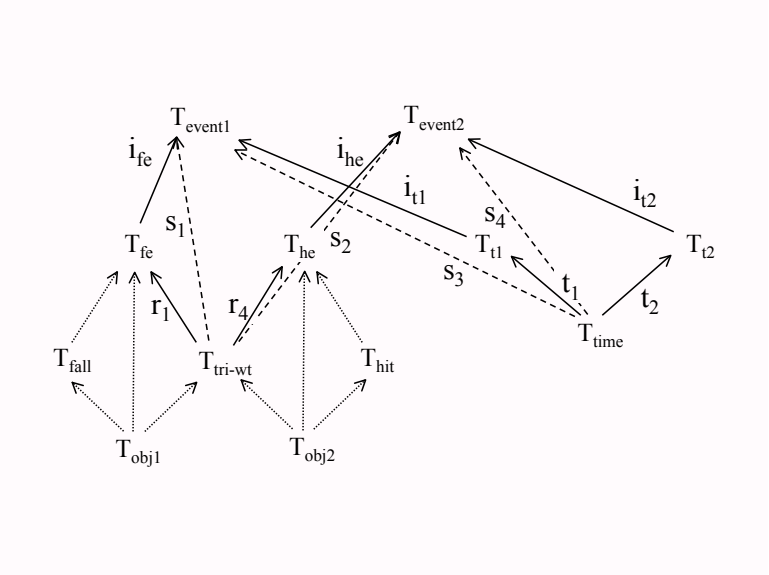


Figure 14: **Morphisms whose domains are the theories $T_{\text{tri-wt}}$ and T_{time}, and whose codomains are T_{event1} and T_{event2}, are formed by composition.**

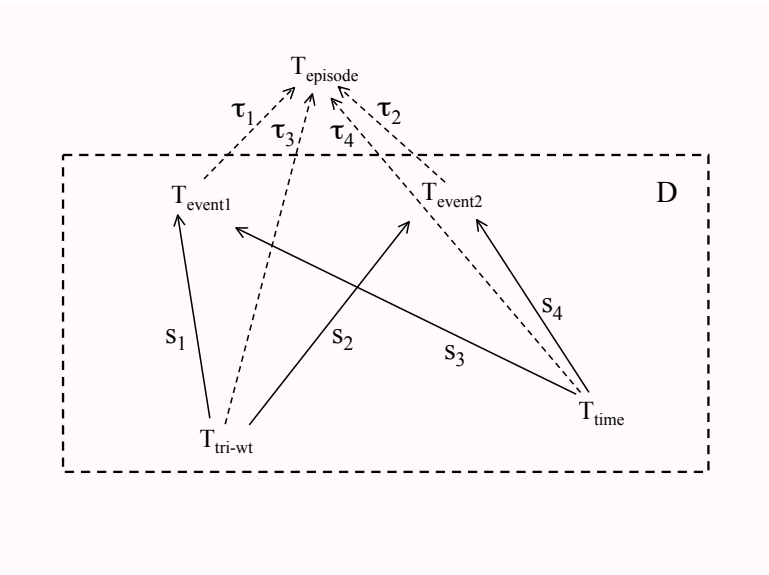


Figure 15: **A final colimit forms the episode.**

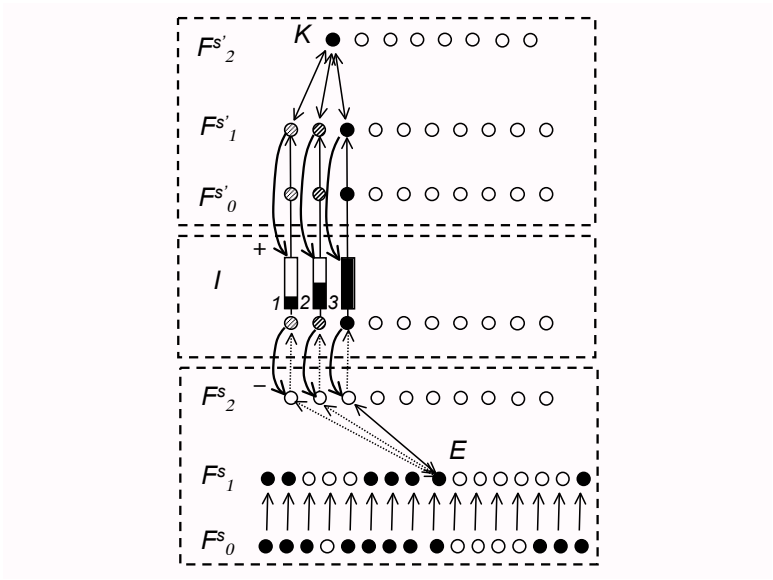


Figure 16: **Initial temporal architecture. The temporal integrator establishes a recency gradient over a sequence of event nodes — F_2 nodes from ART unit s.**

4 A Temporal Colimit Architecture

How can a representation of such a colimit be formed through adaptation in a neural network? To answer this question and to test the CNST in the temporal domain, we have designed an architecture as shown in Fig. 16. It consists of three component networks, designated as units s, I, and s', in a multilevel array that is meant to implement the temporal replay of Figs. 8–10 in Section 3.2. It is also meant to adaptively create the necessary temporal morphisms through the incremental formation of colimits as described in Section 3.3. The feedforward or bottom-up flow from the input event patterns (bottom layer of nodes) to the temporal colimit nodes (top layer) organizes temporal sequences of events in the network memory. The top-down flow provides recall, replaying a temporal sequence in stepwise fashion. The foregoing is merely an overview, for feedback through top-down connections is involved in memory storage and feedback through bottom-up connections is involved in replay.

Units s and s' in Fig. 16 are a modified version of "Fuzzy ART" networks [2]. Only the ART network detail necessary to follow the operational description is shown. The ART units have been modified by omitting the complement part of their input fields, thereby eliminating half of their F_0 and F_1 nodes. Thus, each component of each input pattern, having a "fuzzy" x value scaled to the unit interval, $0 \leq x \leq 1$, has no accompanying "complementary" component with value $1 - x$ as it would in the usual usage of "Fuzzy ART". This omission is primarily to simplify the experiments performed with this initial temporal architecture; that it changes the properties of the templates that form (to be described) is of no concern here.

"Fuzzy ART" is one in a series of Adaptive Resonance Theory (ART) architectures; its purpose is to provide a means of performing a "fuzzy classification" of input patterns with analog or grey-scale component values. Here, we regard it simply as a convenient off-the-shelf architecture for forming the concept representations we need by classifying grey-scale input patterns. The pattern components are scaled to within the unit interval as mentioned, and the scaled values of an input pattern appear as stimulus values which are output by the F_0 (input) nodes of the ART unit. Through one-to-one feedforward connections, these values appear at the F_1 nodes. A pattern-matching operation follows through an interplay of feedforward and feedback activity between the F_1 layer and a mutually competitive layer of F_2 nodes. A winning node $F_{2,k}\,(1 \leq k \leq n_2)$, where n_2 is the number of nodes in the F_2

layer, emerges whose pattern of nonzero weights in its feedforward afferent connections from F_1 best matches the nonzero components of the input pattern. The same pattern of nonzero weights is contained in its top-down connections to F_1 , which form the current *template pattern* Q_k for the input pattern class k associated with $F_{2,k}$. Feedback to F_1 through these connections results in a modified F_1 pattern $F_1 \wedge Q_k$, a pattern of logical "ANDs" with values $\min(F_{1,i}, Q_{k,i})\,(i = 1, \ldots, n_1)$, where n_1 is the number of node in the F_1 layer, which is the same as the number n_0 of its input nodes in F_0. During the pattern-matching operation, this ANDed pattern is tested against the input pattern by a *vigilance subsystem* in the network; if there is not too much "erosion" of the input pattern values via the minimum operation (not too many 1s replaced by 0s, for example), $F_{2,k}$ is said to be in *resonance* with the input, activity in $F_{2,k}$ and the nodes of $F_0 \wedge Q_k$ persists for a brief interval, and the template Q_k undergoes weight adaptation to become $F_0 \wedge Q_k$ (and the feedforward weights from F_1 to $F_{2,k}$ are similarly modified). The input has become the most recent input pattern in the "cluster" of patterns represented by $F_{2,k}$ and its template has been modified accordingly. If, on the other hand, the $F_0 \wedge Q_k$ pattern expresses too great a loss in overall magnitude from the input pattern F_0, an F_2 reset occurs, a new F_2 node emerges from the competition as the winner, it reads out its template over F_1, and the match process is re-enacted. The net effect of the entire process is that the input patterns form "clusters" by virtue of the fact that each one becomes associated with an F_2 node. Because each F_2 node acquires through adaptation a feedback weight pattern Q_k to F_1 that serves as a basis for future input-pattern-matching, it has a semantic representation, a "reason why" input patterns are in its cluster. This justifies the transition in terminology from "cluster" to "class". The usual convention in ART network simulations is for the F_2 nodes to adopt the initial members of their clusters/classes in the order $F_{2,1}, F_{2,2}, F_{2,3}, \ldots$.

We first describe the temporal colimit formation process as it corresponds to the bottom-up flow. There are three stages of processing of the input patterns representing the events, which are sampled at the input layer F_0^s of s. Each event is represented as an ART colimit by a node $F_{2,J}^s\,(1 \le J \le n_{s,2})$ via the ART classification process. Node $F_{2,J}^s$ forwards its output to an I node I_J, which begins to "time-stamp" it by "integrating down". In the next time step, a different F_2^s node will normally become active in place of $F_{2,J}^s$ and the input to I_J will cease; if the input from $F_{2,J}^s$ persists, however, the down-integration process will start over. In any case, the value $1 - I_J$ (where we let the symbol for a node such as I_J denote its current output) represents the elapsed time since the $F_{2,J}^s$ event last occurred, on a scale of 0 to 1. The output of I_J is forwarded to node $F_{0,J}^{s'}$ as input to ART unit s', and in turn is forwarded to node $F_{1,J}^{s'}$. Notice that the number of nodes in these layers is one and the same, $n_{s',1} = n_{s',0} = n_I = n_{s,2}$; this is indicated in Fig. 16.

As node I_J "integrates down" over time its output is continually forwarded to node $F_{0,J}^{s'}$. When an $F_2^{s'}$ node resonates with an input pattern over $F_1^{s'}$, therefore, it is adopting a recency gradient of integrated outputs $I_j\,(1 \le j \le n_{s',1})$ showing the elapsed time since each $F_{2,j}^s$ was last active (with 0 indicating an elapsed time too great to consider); its template will be modified accordingly. The recency gradients, which are continually formed as new events are input at F_0^s, are determined by the following update equations for the integrator nodes I_j at each time step of the simulation:

$$\begin{aligned} \delta_j &= x_j - I_{j,\text{old}}, \\ I_{j,\text{new}} &= \begin{cases} x_j, & \delta_j \ge 0, \\ I_{j,\text{old}} \cdot d_1, & \delta_j < 0 \text{ and } I_{j,\text{old}} \cdot d_1 > I_{\min}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

Here, x_j is the current input to integrator node I_j, $I_{j,\text{old}}$ and $I_{j,\text{new}}$ are the old and new outputs for integrator node I_j, d_1 is the down-time-constant for activity decay in node I_j, and $I_{\min}$ is a "noise" threshold for this activity: at or below this value, the output of I_j is set to zero. From equation (1), each node I_j generates a sequence of decreasing, graded output values over successive time steps except when it receives an input stimulus which momentarily boosts its activity to the stimulus level. We call this process a decay sequence. It results in a recency gradient over I at each time step which either is learned as a template for a newly-active $F_2^{s'}$ node or is used to update the template of a previously-active $F_2^{s'}$ node, depending upon the current winner in the winner-take-all competiton of the $F_2^{s'}$ layer.

The $F_{s',2}$ nodes are to form the colimit representations of the temporal sequences via the ART classification of their inputs. However, there are two issues to consider in relation to this. First, to properly form a temporal colimit representation would in general require a base diagram whose objects and morphisms represent complex information resulting from several previous levels of colimit formation by the neural network, as discussed in Section 3.3 (see Figs. 11–15). In order to simplify the experiments to be performed, the present test architecture forms colimits in only two stages, by harnessing the classification capability of the two ART units. This architecture can be extended to multi-stage colimit formation by including more ART and possibly integrator modules, and so the restriction to two stages is not serious. A greater difficulty is posed by the fact that the colimits formed by the ART architecture have a conceptual difficulty arising from their overly-simple form. In the CNST analysis, each node $F_{2,k}\,(1 \leq k \leq n_2)$ is the carrier of a colimit object whose base diagram consists of neural objects $(F_{1,i}, \eta_i)\,(1 \leq i \leq n_1)$. This diagram is discrete, however, for it contains only a disconnected collection of objects (and their identity morphisms, which are not shown). This means that the ART unit colimits are merely coproducts. As discussed in Section 2.2, coproducts are generally inadequate for expressing complex concepts in terms of simpler concepts because of the lack of blending objects and the requisite morphisms. Again invoking the prerogative of simplicity, we can accept this inadequacy in the colimits formed from the event input patterns by ART Unit s, since the content of these lower-level colimits is not the focus of the experiments to be performed. This is not acceptable, however, for the colimit formation process for an episode as shown in Fig. 11. In general, the integrator-time-stamped F_2^s events must be blended along their common input features, that is, along F_1^s concept representations shared by their templates. In particular, under the assumption that a temporal sequence such as an episodic memory sequence must have a notion of continuity, the lower-level blending objects indicated in the diagram for the entire episode together with the indicated morphisms must be present. These serve to unify the sequence into a meaningful whole based upon information that is shared across the events. This is discussed further in the context of the supertemplate architecture.

4.1 Supertemplates: Augmenting the Compositions

Now, we contrast the temporal colimit architecture of Fig. 16 with that of Fig. 17. For simplicity in referring to the neural category items, let us eliminate intervals η in what would have been, for example, $(F_{1,1}^s, \eta_1)$, and simply let $F_{1,1}^s$ denote the object; that is, η_1 includes all nonzero outputs of $F_{1,1}^s$. However, we shall retain the intervals in reference to Figs. 8–10. Also, let us eliminate the names of the connections in the paths in Figs. 16 and 17, which are anyway unlabelled, and use only the nodes to characterize the connection paths (signal paths) in these figures. Finally, let us omit the terminology "carrier", which distinguishes nodes and signal paths in a neural architecture from objects and morphisms in the corresponding neural category, and refer to objects and morphisms as being "active" or "activated", "[an object] having afferent connections", etc. This will provide a notational shorthand and improve the readability of the following discussion.

There is a conceptual problem with the architecture of Fig. 16 and, as will be shown, a performance issue is associated with this. The conceptual problem is that the architecture does not provide the unique signal path that distinguishes a temporal neural morphism from other neural morphisms having the separate paths, which are included. There is no way to disambiguate a collection of "time-stamped events" (the ART unit s' recency gradient components represented by top-down template connections to $F_1^{s'}$) from a *sequence* of time-stamped events. Regardless of the fact that they are "time-stamped" by the integrator when an ART template is formed, there is nothing to ensure that the events over an entire episode have anything in common, which would be represented by persistently-active F_1^s nodes. In an episodic memory sequence representing one person's memory of a meeting with another person, for example, the persistence of an F_1^s node might represent the other person's face or some other aspect of their presence; that this contextual information is present throughout the meeting could be the major determinant for the individual's brain forming the episode. Such persistently-active nodes correspond to the node p_0 in the temporal morphism architecture of Figs. 8–10. In the latter, inhibitory feedback from nodes p_3 and p_4 to nodes p_1 and p_2 suppresses the activation of paths γ_1 and γ_2 except when their times for activation occur during replay. The appropriate time occurs when the activity of the associated node p_3 (p_4) first declines to a level at which it can no longer suppress p_1 (p_2). But in order for p_1 (p_2) to become active, the

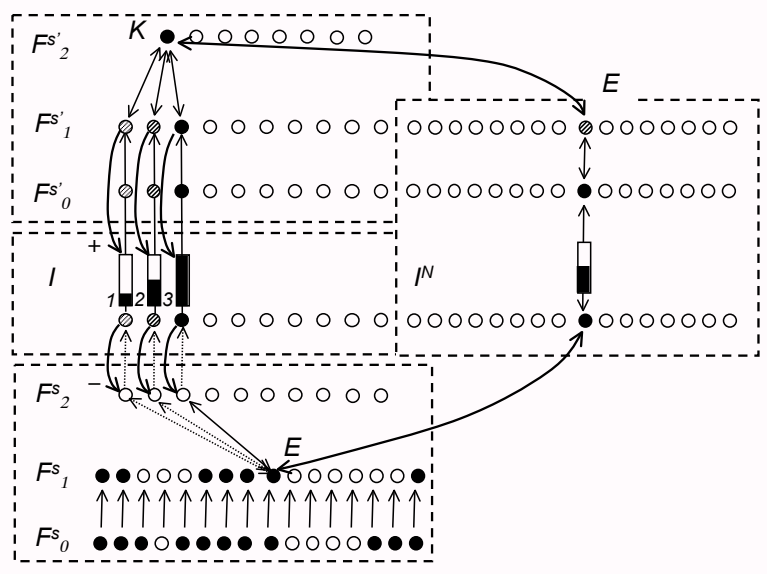


Figure 17: **Initial temporal architecture with an ART unit s' supertemplate connection path to F_1 node E of ART unit s. The nonlinear temporal integrator I^N supplies an intermediate node for this path.**

release of inhibition is not enough; it requires excitatory input. This is supplied by node p_0, which is supposed to remain fully active throughout the replay of the sequence. This is assured by the excitation supplied by the colimit carrier node p_5 through the top-down connection c_{-7}, the reciprocal connection opposite the bottom-up signal path $[(p_0, \eta_0), c_7, (p_5, \eta_5)]$ associated with the colimit leg morphism $M(\tau_3)$. But in Fig. 16, the only source of excitatory input to F_2^s when an $F_2^{s'}$ node is active but there is no bottom-up input to F_1^s via F_0^s is via the top-down connections from $F_2^{s'}$ to $F_1^{s'}$, from $F_1^{s'}$ to I, and from I to F_2^s. As will be seen, this is problematic not only for the continuity of a temporal sequence because there is no unique path connecting an $F_2^{s'}$ node with the persistent F_1^s nodes unifying its events, but also for the activation of the F_2^s nodes to effect the replay of the events in its sequence. The reason for the latter problem is that the top-down connections from I to F_2^s must be inhibitory, as discussed for the scheme of Figs. 8–10. That is, the nodes in F_2^s play the same role as p_1, p_2, the integrator nodes play the role of p_3, p_4, and $F_{1,E}^s$ plays the role of p_0. That $F_{1,E}^s$ has no obvious source of excitation during replay (when a bottom-up stimulus may not be present), as p_0 must have, is the operational dilemma, which is a reflection of the conceptual difficulty of the missing information at $F_2^{s'}$ about the persistence throughout the sequence of the activity of $F_{1,E}^s$. The F_1^s nodes that are active throughout a temporal sequence are the domains of the temporal morphisms. Under our proposal for temporal colimits in Section 3.2, the diagram object associated with each of these persistently-active F_1^s nodes must be accompanied by a unique temporal morphism signal path in the architecture. In particular, this requires a connection path with reciprocal between $F_{1,E}^s$ and the appropriate $F_2^{s'}$ node in addition to those which pass through F_2^s and the linear integrator. As shown for $F_{1,E}^s$ in Fig. 17, a newly-added path and its reciprocal, corresponding to the reciprocal path $[(p_5, \eta_5), c_{-7}, (p_0, \eta_0)]$ in Figs. 8–10, provides a source that enables the necessary excitation to maintain its activity throughout the replay. This modification to the architecture, illustrated in Fig. 17, results in $F_2^{s'}$ templates which effectively extend over *both* ART units. We call this kind of structure a *supertemplate*.

More detail is provided by the following operational description of the architecture. A temporal colimit object — one is shown at the top of Figs. 16 and 17, $F_{2,K}^{s'}$ — corresponds to the object (p_5, η_5) in Figs. 8–10, and objects $F_{1,i}^{s'}$ to which it has nonzero template connections $Q_{K,i}^{s'}$ play the role of the objects (p_3, η_3) and (p_4, η_4). A morphism in the base diagram of the colimit is associated with a path $[F_{1,E}^s, F_{2,J}^s, I_J F_{0,J}^{s'}, F_{1,J}^{s'}]$. Fig. 17 highlights

three such paths, although they are also present (but somewhat less obvious) in Fig. 16. During the bottom-up activation of the base diagram, an input at $F^s_{1,E}$, which is part of a diagram in ART unit s having colimit object $F^s_{2,J}$, aids in the activation of the latter object. Its output has unit strength, and this activates a temporal integrator node I_J which initially produces the same output. As time goes on, the output of I_J decreases, "integrating down" according to the value of the down-time-constant in equation (1). The output of I_J registers through a bottom-up connection at input node $F^{s'}_{0,J}$ of ART unit s', then at $F^{s'}_{1,J}$. At some point in the process, the usual ART template modification ensues while an $F_2^{s'}$ node $F^{s'}_{2,K}$ is active. If $Q^{s'}_{K,J} \neq 0$, the corresponding feedforward weight is also nonzero, allowing the path $[F^{s'}_{1,J}, F^{s'}_{2,K}]$ to be active, as is also the path $[F^s_{1,E}, F^s_{2,J}, I_J\, F^{s'}_{0,J}, F^{s'}_{1,J}]$. The composition of the morphisms associated with these two paths yields the morphism associated with the concatenated path $[F^s_{1,E}, F^s_{2,J}, I_J, F^{s'}_{0,J}, F^{s'}_{1,J}, F^{s'}_{2,K}]$. This corresponds to the composition $M(\tau_1) \circ M(s_1)$ associated with the concatenation of paths $\gamma_1; [(p_3, \eta_3), c_5, (p_5, \eta_5)]$ in Figs. 8–10, which combines a base diagram morphism with a cocone leg morphism. Another composite path active at a later time but while the activities of both $F^s_{1,E}$ and $F^{s'}_{2,K}$ persist corresponds to the composition $M(\tau_2) \circ M(s_2)$ associated with the path $\gamma_2; [(p_4, \eta_4), c_6, (p_5, \eta_5)]$. Just for completeness, let the second path be $[F^s_{1,E}, F^s_{2,L}, I_L, F^{s'}_{0,L}, F^{s'}_{1,L}, F^{s'}_{2,K}]$. Given that the constraints on a temporal morphism are satisfied, the four morphisms involved in these two compositions form a pushout square. Note the heights of the bars in Figs. 16 and 17, forming a recency gradient over integrator nodes labelled "1, 2, 3". We can regard nodes "1" and "2" as I_J and I_L, whose outputs (two components of the recency gradient) register at two vertices $F^{s'}_{1,J}, F^{s'}_{1,L}$ of a pushout square (the vertical bars are not duplicated there to simplify the picture). The inclusion of the third path, containing node "3", results in a total of three pushout squares (the three combinations of two paths through 1, 2, 3) with blending object $F^s_{1,E}$ and pushout object $F^{s'}_{2,K}$. Together, the pushouts make up the colimit defining diagram for a 3-event temporal sequence.

For convenience in operating with the two ART units and the integrator in the architecture of Fig. 17, both the bottom-up and top-down paths, $[F^s_{1,E}, F^{s'}_{2,K}]$ and its reciprocal $[F^{s'}_{2,K}, F^s_{1,E}]$, include more than just these two nodes. They also include intermediate nodes provided by the extensions to the layers $I, F_0^{s'}, F_1^{s'}$ which are shown. These facilitate the adaptive formation of a strong supertemplate connection between $F^{s'}_{2,K}$ and $F^s_{1,E}$ during the initial formation of the supertemplate $Q^{s'}_K$ as a result of $F^s_{1,E}$ having been persistently active for a sufficient number of time steps as determined by the nonlinear integrator values $u_{\rm nl}, d_{\rm nI}$ and θ^I_0 in the following equations, where x_j is the current input to integrator node I^N_j, $\theta^I_{j,\rm old}$ and $\theta^I_{j,\rm new}$ are the (internal) activation values for I^N_j before and after the update, θ^I_0 is the uniform threshold value for the nonlinear integrator layer I^N, and ϕ_I is its uniform signal function. The up-time-constant for increasing activation for each I^N_j at each time step is $u_{\rm nl}$; the down-time-constant for activity decay is $d_{\rm nI}$. As can be seen, the I^N nodes are binary with uniform output value $w^I > 0$ when their activation values exceed the uniform threshold. The equations are

$$\begin{aligned} \delta_j &= x_j - \theta^I_{j,\rm old}, \\ \theta^I_{j,\rm new} &= \begin{cases} \theta^I_{j,\rm old} + u_{\rm nl} \cdot \delta_j, & \delta \geq 0, \\ \theta^I_{j,\rm old} + d_{\rm nl} \cdot \delta_j, & \text{otherwise}, \end{cases} \\ I^{\rm new}_j &= \phi_I(\theta^I_{j,\rm new} - \theta^I_0) \end{aligned} \tag{2}$$

where

$$\phi_I(z) = \begin{cases} w^I, & z > 0, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

When $F^s_{1,E}$ has been persistently active for a number of time steps as determined by equations (2)–(3), θ^I_E reaches a value such that $\theta^I_E - \theta^I_0 > 0$ and as a result $\phi_I(\theta^I_E - \theta^I_0) = w^I$. This value is forwarded to $F^{s'}_{0,E+n_{s',1}}$, then $F^{s'}_{1,E+n_{s',1}}$. During the usual ART pattern-matching operation, resonance, and subsequent weight adaptation, when $F^{s'}_{2,K}$ is a newly-committed node the weight in the top-down connection $[F^{s'}_{2,K}, F^{s'}_{1,E+n_{s',1}}]$ will adapt to the value w^I and that in the bottom-up connection $[F^{s'}_{1,E+n_{s',1}}, F^{s'}_{2,K}]$ will undergo the corresponding ART bottom-up adaptation. The other connections in the top-down and bottom-up paths have unit weights, and, hence, the

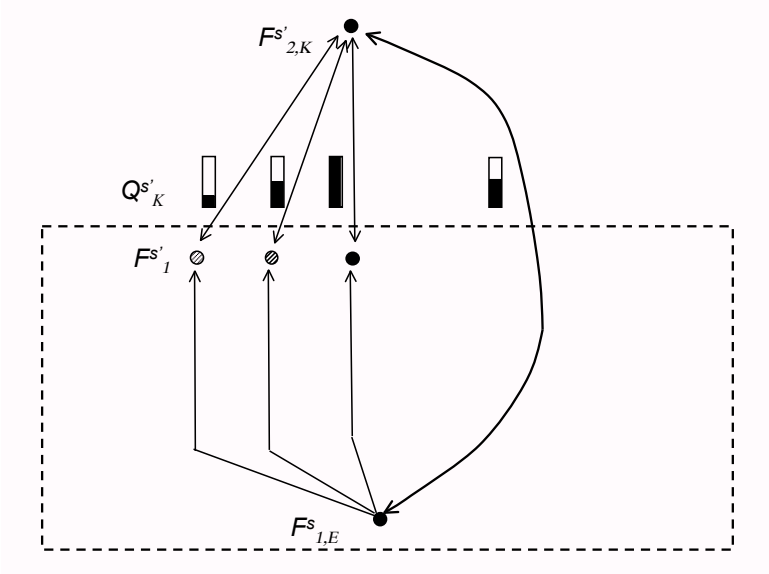


Figure 18: **A supertemplate representing the defining diagram of a temporal colimit. The supertemplate weights $Q^{s'}_{K,i}$ are shown as vertical bars, with three recency gradient components on the left and a weight derived from the nonlinear integrator on the right.**

node $F^{s}_{1,E}$ has effectively been connected to $F^{s'}_{2,K}$ with the template weight value w^I, supplementing the recency gradient template connections $[F^{s'}_{2,K}, F^{s'}_{1,i}]$ in the newly-formed supertemplate $Q^{s'}_{K}$ (refer to Fig. 17). The new supertemplate has component weight values $Q^{s'}_{K,i} = I_i\,(i = 1, \ldots n_{s',1})$ and

$$Q^{s'}_{K,n_{s',1}+i} = \begin{cases} w^I, & F^{s}_{1,i}\,\text{persistent}, \\ 0, & \text{otherwise} \end{cases} \qquad (i = 1, \ldots n_{s,1}).$$

There is a further operational difficulty in Fig. 16 in addition to that of replaying a temporal sequence, again a reflection of the conceptual difficulty in the lack of supertemplate connections. This occurs in the learning of temporal sequences. The lack of unique signal paths in the path sets defining temporal morphisms makes it difficult for the architecture to distinguish between sequences which have mostly the same events, but have significant differences in their persistent F^{s}_{1} nodes. This will be illustrated in the following sections, which recount two experiments that test the effectiveness of the CNST modeling of event sequences presented here. The two versions of the architecture in Figs. 16 and 17 provide the vehicle for the tests. The first experiment differentiates between the presence and absence of supertemplates in learning a sequence of input events through connection-weight adaptation. The second experiment tests the ability of an adapted network with supertemplates to replay a sequence.

5 Experiment: Learning Temporal Sequences

We conducted an experiment to test the effect of a supertemplate connection upon the learning of temporal sequences in the architecture of Fig. 17 compared with that of Fig. 16, which does not have supertemplates. The hypothesis is that the supertemplate architecture, with the F^{s}_{1} blending objects and their unique connection paths to/from the $F^{s'}_{2}$ nodes, can distinguish between different event sequences having similar recency gradients

in cases for which the architecture without supertemplates cannot. Because the architecture of Figure 16 has no supertemplate connections between F_1^s and $F_2^{s'}$, it has only the recency gradients to guide the classification of sequences of ART unit s input patterns by ART unit s'. As a consequence, a low enough vigilance value in ART unit s' could allow enough generalization for two temporal sequences with a mismatch in only one or two of their events to appear indistinguishable. By contrast, in the architecture of Fig. 17 the ART s' classification is supplemented by the supertemplate connections to the temporal colimit defining diagram blending objects such as $F_{1,E}^s$ in the previous discussion. If the dissimilar events in the same two temporal sequences result in F_1^s patterns whose *persistently-active* nodes differ substantially, the two sequences will share few or no supertemplate connections. This results in a loss of similarity in the two sequences in comparison with their similarity in the architecture without supertemplates, which is based solely upon the two recency gradients. This in turn will affect the ART $F_2^{s'}$ choice and vigilance operations, which determine whether or not the two sequences are classified the same.

The bulk of the experiment consisted of an analysis based upon a knowledge of ART classification and an analysis based upon the linear and nonlinear temporal integrator equations (1)–(3) to select input pattern sequences and parameter settings for the architectures of Figs. 16 and 17. The simulations that followed were then a demonstration that inputs and parameter settings exist for which the hypothesis can be confirmed.

For both the supertemplate and no-supertemplate architectures, the vigilance value ρ_s of ART unit s was set to $\rho_s = 0.9$ in this experiment, a value high enough that each distinct event input pattern (to be shown) forms a separate F_2^s class. Therefore, the connection-weight templates that are encoded by ART unit s duplicate its input patterns. This simplifies the experiment by allowing the encoding of templates by multiple patterns only in ART unit s', restricting the focus to generalization over sequences of input events by eliminating the effects of generalization in the events themselves. On the other hand, $\rho_{s'}$ was set to $\rho_{s'} = 0.75$, a value which allows for the encoding of templates by multiple recency gradient patterns, facilitating generalization over sequences.

The "down" time constant $d_{\rm I}$ for the linear integrator in equation (1) was set to $d_{\rm I} = 0.6$. When an F_2^s node $F_{2,J}^s$ first forwards its output of unity to I_J, the latter registers an activation value (= output value) of $I_J(t) = 1.0$. The inputs were such that each successive input pattern differed from the previous one, so that a different F_2^s node became active with each step and therefore $x_J = 0$. Thereafter, the activity level of $I_J(t)$ would change according to $I_J(t+1) = 0.6 \cdot I_J(t)$ at each time step, where $I_J(t+1)$ is the quantity $I_{J,\rm new}$ in (1). Since I is a layer of linear nodes, the output of I_j is the same as its activity level unless the latter falls below the "noise" threshold $I_{\rm min}$, in which case it is zero. In this experiment $I_{\rm min}$ was set to 0.1. The output of I_j is continually relayed to unit s' input node $F_{0,J}^{s'}$, so that at each time step the $F_0^{s'}$ layer registers the current recency gradient. With the down-time-constant set to 0.6 and with a "noise threshold" of 0.1, a maximum of five $F_{0,j}^{s'}$ nodes were active at any time due to activity decay. Their outputs duplicated the above-threshold portion of the current recency gradient over the nodes I_j.

For the supertemplate architecture, all parameters shared with the no-supertemplate architecture were set to the same values. The "up"/"down" time constants $u_{\rm nl}$ and $d_{\rm nl}$ for the nonlinear integrator nodes, which are intermediaries in the supertemplate signal paths, were set to the values $u_{\rm nl} = 0.3$ and $d_{\rm nl} = 1.0$ in (2)–(3). The threshold value for the nonlinear integrator nodes I_j^N was set as $\theta_0^I = 0.8$, and for $z = \theta_{j,\rm new}^I - \theta_0^I > 0$ the signal function output was set to $\phi_I(z) = w^I = 0.5$. This had the effect of requiring that in order for a supertemplate connection path J to become active, $F_{1,J}^s$ must be persistently active for at least five successive events, in which case the intermediate supertemplate connection path node $F_{1,J}^{s'}$ would output the value 0.5. This value is forwarded to the current winner-take-all node $F_{2,K}^{s'}$ and, following resonance, becomes its supertemplate weight for the top-down connection to $F_{1,J}^{s'}$. As previously explained, the connections from/to F_1^s to/from I^N and those from/to I^N to/from $F_1^{s'}$ are fixed at the value 1.0. This leaves the ART reciprocal connections between $F_1^{s'}$ and $F_2^{s'}$ as the only adaptive connections in the supertemplate connection paths. Their weights extend the ART unit s' templates beyond the original templates stored in the connections between $F_1^{s'}$ and $F_2^{s'}$. These supertemplate weights, like the original template weights, have initial values of 1.0 and, after encoding, are all 0.0 except for those values corresponding to the nonzeros in a bottom-up/top-down pattern match (the input

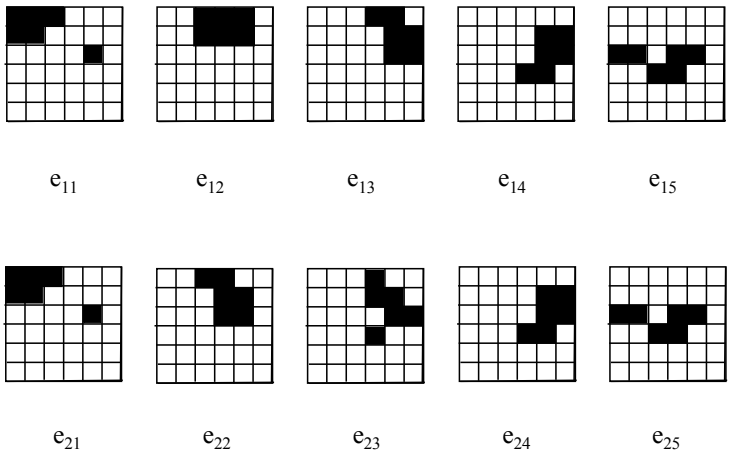


Figure 19: **The events in sequences e_1 (top) and e_2 (bottom). Notice that pixel $(5,3)$ is active (black) in all 5 images of e_2, but is inactive in image 2 of e_1.**

pattern/template pattern AND operation described earlier). In total, the supertemplate of an $F_2^{s'}$ node encodes that part of the superimposed recency gradients for the temporal sequences it adopts and their associated persistent F_1^s active nodes that survive all of the pattern-match operations at $F_1^{s'}$.

The experimental hypothesis was tested by supplying the same two sequences of five events each, first e_1 and then e_2, to both architectures exactly as shown in Fig. 19, with the full 10-pattern sequence repeated once. For simplicity, e_2 has a single input pattern pixel that is ON in all five input patterns and e_1 does not have such a pixel. This results in a substantial difference in the supertemplates for the two sequences, with e_2 having a single blending object and with e_1 having none. The input patterns are binary, with white $= 0$ and black $= 1$ in Fig. 19. They are denoted $e_{i,j}\,(i = 1, 2;\ j = 1, \ldots, 5)$, presented successively to ART unit s in an input field of $6 \times 6 = 36$ binary pixels at the F_0^s layer. Notice that the single pixel $(3,5)$ (row 3, column 5) is present in all 5 events of e_2 and in events $e_{1,1}, e_{1,3}, e_{1,4}, e_{1,5}$, but not in event $e_{1,2}$, of e_1. Thus, within a certain range of nonlinear integrator up/down time constants including the values stated in the preceding paragraph the pixel in row 3, column 5 represents a blending object for the 5 events as objects in the defining diagram of a colimit for e_2. This does not hold for e_1 because of the lack of reinforcement in the $e_{1,2}$ input pattern.

5.1 Results

Because the value $\rho_s = 0.9$ is sufficently high, the template patterns $Q_1^s, \ldots, Q_5^s$ that form exactly match the input patterns $e_{1,1}, \ldots, e_{1,5}$ from which they were encoded, as expected. Subsequently, $e_{2,1}, \ldots, e_{2,5}$ are input; three of these, $e_{2,1}, e_{2,4}$ and $e_{2,5}$, exactly match the three templates Q_1^s, Q_4^s and Q_5^s and therefore are adopted by the same nodes $F_{2,1}^s, F_{2,4}^s$ and $F_{2,5}^s$ as were $e_{1,1}, e_{1,4}$ and $e_{1,5}$, respectively. However, because of their dissimilarity with the corresponding e_1 patterns, the two input patterns $e_{2,2}, e_{2,3}$ encode new templates, Q_6^s and Q_7^s for nodes $F_{2,6}^s$ and $F_{2,7}^s$. Figs. 20– 43 in Appendix A show the graphic displays for two successive passes through both sequences for the architecture of Fig. 16, then that of Fig. 17. Each input event is shown as a 6×6 pixel pattern displayed at the bottom. The template following coding/recoding for the current input is shown just above the input pattern, and the activity of the corresponding F_2^s node is shown in the bar above that, which shows the entire

F_2^s array (20 nodes were present, an ample provision for potential templates). The output (which equals unity) of the currently-active F_2^s node is forwarded to the corresponding integrator node, shown in the next bar upward in the graphics; this bar is labelled "Temp Int T0". Each integrator node initially generates a unit output but begins to decrease in activity when its input stimulus is removed as previously described based upon equations (1) for the linear integrator layer I and (1)–(3) for the nonlinear integrator layer I^N. A recency gradient is available as the output of the linear integrator nodes at each time step. Initially, the recency gradients have fewer than 5 components as they gradually build to the full 5 as $e_{1,1}, \ldots, e_{1,5}$ are input successively, finally resulting in a full 5-component gradient with $e_{1,5}$ (Fig. 24). With each step, the integrator outputs are the inputs to the corresponding $F_0^{s'}$ nodes, then to $F_1^{s'}$ (shown at successively higher levels in the graphics). Following the ART pattern-matching and resonance, a node $F_{2,K}^{s'}$ adopts the current recency gradient into its input class. Its template $Q_K^{s'}$ (the horizontal bar labelled "SP Res Temp (value K)" for "s' resonant template K ") either is encoded as a recency gradient or is recoded to generalize over multiple recency gradients.

Without supertemplates

During the first presentation of e_1 a separate recency gradient template was encoded with each time step because they were dissimilar with respect to the ART matching criterion with the vigilance value used ($\rho_{s'} = 0.75$). This is a reflection of the fact that although the full gradients for the two input sequences e_1, e_2 are the focus of the experiment, the special status given to e_1 and e_2 is simply an experimental convenience. The network continuously encodes recency gradients, beginning a new $F_2^{s'}$ template when the output of I at the current step differs significantly from that at the previous step according to the ART network similarity criteria. Figs. 20–31 in Appendix A show the simulation graphically for the no-supertemplate architecture of Fig. 16. The simulation begins with event $e_{1,1}$ at time step 1 (Fig. 20); the initial template $Q_1^{s'}$ has $Q_{1,1}^{s'} = 1.0$ (black) and $Q_{1,j}^{s'} = 0.0$ for $j > 1$ (white). Event $e_{1,2}$ at time step 2 yields (Fig. 21) $Q_{2,1}^{s'} = 0.6$ (darkest grey), $Q_{2,2}^{s'} = 1.0$ (black), and $Q_{2,j}^{s'} = 0.0$ for $j > 1$ (again white). This process continues, and the fifth template $Q_5^{s'}$ registers the recency gradient for the full five-step sequence e_1 (Fig. 24) as $[0.13, 0.21, 0.36, 0.6, 1.0, 0.0, \ldots, 0.0]$ (note the progressive shades of grey increasing to black from left to right, then all white again). Because $e_{2,1}$ duplicates $e_{1,1}$, it has the same ART unit s template Q_1^s; however, by this time, there is a full recency gradient available as input to ART unit s'. This deviates substantially from $Q_1^{s'}$, which has a single nonzero component. Therefore, although it has the same template as $e_{1,1}$ in unit s, it encodes a new template $Q_6^{s'}$ in unit s', with $Q_6^{s'} = [1.0, 0.13, 0.22, 0.36, 0.6, 0.0, \ldots, 0.0]$ (Fig. 25). This continues with the encoding of $Q_7^{s'}$, $Q_8^{s'}$, and $Q_9^{s'}$. However, the processing of $e_{2,5}$, instead of ending with the encoding of a new template $Q_{10}^{s'}$, incurs the *recoding* of template $Q_5^{s'}$, to produce $Q_5^{s'+} = [0.13, 0.0, 0.0, 0.6, 1.0, 0.0, \ldots, 0.0]$ (Fig. 29). This happens because the recency gradient that is available after $e_{2,5}$ occurs is just similar enough (given $\rho_{s'} = 0.75$) that instead of causing a reset, it resonates with the existing template $Q_5^{s'}$. As a result, components $Q_{5,2}^{s'}$ and $Q_{5,3}^{s'}$ are now zero because events 2 and 3 in e_2 are different from those in e_1. All components $Q_{5,i}^{s'}(i > 5)$ were already zero. Since the other three recency gradient components of e_2 exactly match those of $Q_5^{s'}$, they are retained as $Q_{5,1}^{s'}, Q_{5,4}^{s'}$ and $Q_{5,5}^{s'}$.

At this point, the full recency gradients of e_1 and e_2 have been adopted into the same ART unit s' class but with a template that excudes the second and third events of both sequences, where these events are "zeroed out" by the ART pattern AND operation at $F_1^{s'}$. The input of the two sequences was repeated in the same order. On this second pass, the input $e_{1,1}$ (Fig. 30) produced (because of the intervening processing of e_1, e_2) the recency gradient $I = [1.0, 0.0, 0.0, 0.36, 0.6, 0.13, 0.22, 0.0, \ldots, 0.0]$. Notice the values $0.13, 0.22$ for I_6, I_7, which are "integrated down" from the values at the times $e_{2,2}, e_{2,3}$ were first input. This input resulted in the recoding of a template, $Q_6^{s'}$, as $[1.0, 0.0, 0.0, 0.36, 0.6, 0.0, 0.0, 0.0, \ldots, 0.0]$. On the other hand, the second pass of $e_{1,2}$ in the next time step produced a recency gradient which encoded a new template, $Q_{10}^{s'} = [0.6, 1.0, 0.0, 0.22, 0.36, 0.0, 0.13, 0.0, \ldots, 0.0]$ (notice that the recency gradient has advanced one time

step from the second pass of $e_{1,1}$, and the value 0.13 of $Q^{s'}_{10,7}$ retains the recency gradient value at I_7).

The preceding discussion of the simulation with the architecture of Fig. 16 is intended not only to clarify its operation, but also to highlight the ANDing of the full recency gradients from e_1 and e_2 as a consequence of their being adopted into the same ART unit s' class, whose template is $Q^{s'}_5$. In conclusion, in learning, the architecture can place two closely-related event sequences in the same class. Its only degree of freedom in avoiding this is the parameter $\rho_{s'}$, for which an increase in value would be necessary. However, a higher value $\rho_{s'} > 0.75$ would inhibit generalizing over similar recency gradients, which can be a disadvantage if the events that generated the gradients share much the same information and therefore indicate that the similarity is indeed meaningful. But for this similarity to be truly meaningful, the network must be capable of responding differently if the two recency gradients represent event sequences with contexts or continuity information which differs significantly. This information is represented by those input components which the events in a sequence have in common—that is, which have persistent activity throughout a sequence.

With supertemplates

Next, the same experiment with the sequences e_1 and e_2 was run with the supertemplate architecture of Fig. 17. This was to test its capacity for the disambiguation of temporal sequences having similar recency gradients but differing in the information content shared by their events. All of the network parameter values used for the non-supertemplate architecture were the same in this simulation. A new degree of freedom was provided by the newly-added supertemplate connection paths between F^s_1 and $F^{s'}_2$, which contain the nonlinear integrator nodes with the "up" and "down" time constant values $u_{\rm nl} = 0.3$ and $d_{\rm nl} = 1.0$ in (2)–(3), threshold $\theta^I_0 = 0.8$, and signal function output $\phi_I(z) = w^I = 0.5$ when $z = \theta^I_{j,\rm new} - \theta^I_0 > 0$. This had the effect of requiring that in order for a supertemplate j connection to become active, $F^s_{1,j}$ must be persistently active for at least five successive events, in which case it would output the value 0.5; this would become the weight for the adaptive connection from $F^{s'}_2$ to the appropriate node in the extended $F^{s'}_1$ layer, and thereby the weight for the corresponding supertemplate connection path. Figs. 32–43 in Appendix B show the result of performing the same experiment with the two sequences e_1 and e_2, but this time with the supertemplate architecture. The new display is the same as the previous one except that the contribution of the nonlinear temporal integrator is now included along with that of the linear temporal integrator. As opposed to the horizontal bar (Temp Int T0) showing the linear integrator input to $F^{s'}_0$, the nonlinear integrator (NL Temp Int T0) input to the extended $F^{s'}_0$ is shown as a 6 X 6 square, since its nodes express the time-integrated activity of the inputs from F^s_1. The "SP Res Temp" display, in turn, shows the thresholded values of the supertemplate derived from "NL Temp Int T0". These are zero except where an F^s_1 node has been active for at least the preceding 5 inputs.

The simulation proceeded as it did for the first architecture until the supertemplate connection shown in Fig. 17 became active. Notice that all values in the supertemplate square were zero (Figures 32–37) until event $e_{2,2}$ occurred, signifying that the input patterns $e_{1,1}, e_{1,2}, e_{1,3}, e_{1,4}, e_{1,5}, e_{2,1}$ did not yield a sequence containing shared information (because $e_{1,2}$ is missing this information). Thereafter, several five-step sequences with continuity information (shared input component at pixel $(3,5)$) appeared, with the value 0.5 output by the corresponding $F^{s'}_1$ node. Since our focus is upon the full sequences e_1 and e_2, we highlight the occurrence of event $e_{2,5}$. Here (Figure 41) the same persistent input was present (pixel $(3,5)$), indicating that the 5 events in e_2 share this continuity information. As a consequence, as opposed to the outcome of the non-supertemplate simulation, the input pattern $e_{2,5}$ did not resonate with the the supertemplate $Q^{s'}_5$, and, hence, did not recode it. Instead, it encoded a new supertemplate, $Q^{s'}_{10}$.

5.2 Discussion of Results

The separate outcomes of the two simulations of temporal sequence learning are evidence that including the unique supertemplate connection paths in a temporal neural morphism provides the extra degree of freedom needed to disambiguate sequences of events having similar recency gradients (but with one or more differing

events) while sharing different information across their events. The new supertemplate paths are the persistently-active paths for a temporal neural morphism, a notion which arose in expressing a sequence of events as a concept colimit. In the experiment, the theory (concept) of a sequence of events is the apical object in a colimit cocone. This is part of the defining diagram for the colimit, which includes not only the separate theories expressing the events, but also blending objects and morphisms which express the information that is shared by all events in the sequence. This structure can be mapped as a functorial image into a neural category expressing a properly-designed neural network. The results presented here suggest that this categorical formalization of temporal sequence learning is useful not only in understanding the acquisition of temporal concept representations, but also in designing networks with greater discriminative power in this form of learning.

5.3 Temporal Replay via Supertemplates

A simulation was performed to demonstrate that the supertemplate ART/temporal-integrator architecture can replay a representation of a temporal sequence of events in a stepwise fashion, where the sequence has been learned as a colimit. The sequence learned by the neural network of Fig. 17 is displayed in the simulation steps shown in Figs. 44–48. The graphic output for the 5 time steps of recall for the sequence e_2, captured in the supertemplate $Q_{10}^{s'}$, is shown in Figs. 44–48 in Appendix C. The temporal replay mode of the simulator is separate from the learning mode. Based upon studies of memory in cognitive and neural science, in a full, biologically realistic simulation the recall mode for a previously-learned event sequence would occur simultaneously with further learning, thereby potentially modifying the sequence. To simplify the study of the temporal colimit morphisms, the present simulator software enacts the two modes separately. Later simulations will investigate the combined learning and replay modes.

During recall, an $F_2^{s'}$ node—$F_{2,K}^{s'}$, using the notation from our earlier example —receives a stimulus. This could occur through the bottom-up connections of the supertemplate architecture indicated in Fig. 17, but then would involve the learning mode together with replay. At present, we are concerned with an independent recall scenario involving a separate stimulus input to $F_{2,K}^{s'}$. The source of the stimulus would be in another part of a larger neural network within which the supertemplate architecture is embedded; for example, the recall (and replay) of an episodic memory in this way would involve cognitive input from a different brain region such as the prefrontal cortex. The separate-stimulus scenario was suggested in the discussion of the temporal neural morphism of Figs. 8–10, where a separate excitatory input to the codomain node of the morphism is shown; here, the codomain node is the colimit $F_{2,K}^{s'}$ node. The stimulus excites $F_{2,K}^{s'}$, which then excites $F_{1,E}^{s}$ via the adaptively-learned, strong supertemplate connection. In Figs. 8–10 this connection is c_{-7}; in Fig. 17, it is the reciprocal path from $F_{2,K}^{s'}$ to $F_{1,E}^{s}$; in the replay simulation figures in Appendix C, $F_{1,E}^{s}$ is the node $(3,5)$ in the 6 X 6 input array. The excitation of $F_{1,E}^{s}$ results in signals from it to the F_2^s nodes to which it has nonzero bottom-up and corresponding top-down template connections, initially resulting in the activation of these nodes and initiating the usual ART F_2 competition among them. However, the $F_1^{s'}$ nodes with template connections from $F_{2,K}^{s'}$ have become active, registering the recency gradient of the temporal sequence it represents. Each of these nodes provides excitatory input through the top-down connections to their correspondents in the I integrator layer, causing the recency gradient to register there. Through the top-down inhibitory connections from the I nodes to their F_2^s correspondents, they suppress them, cancelling the effect of the bottom-up signals from $F_{1,E}^{s}$. Let us assume for the present that the F_2^s nodes having $F_{1,E}^{s}$ in their nonzero template connections are just those receiving top-down inhibition from the current recency gradient in I. This ensures that the entire F_2^s layer is shut down by the inhibition, for in general $F_{1,E}^{s}$ could have nonzero template connections with other F_2^s nodes as well. The removal of this assumption will be addressed presently.

Eventually, the first recency gradient component in I (the node with the smallest activity, number 1 in Fig. 17) "integrates down" to below the "noise" threshold; as a consequence, it releases its inhibition of its F_2^s node. Because $F_{1,E}^{s}$ has been persistently active, having received a continuing stimulus via the $F_{2,K}^{s'}$ to $F_{1,E}^{s}$ connection path, the just-released F_2^s node, still receiving the stimulus from $F_{1,E}^{s}$, again becomes active; now, with no competition and no inhibitory input, it displays its template over F_1^s. The first event in the sequence, $e_{2,1}$, is now undergoing replay (Fig. 44 in Appendix C). As in the learning mode simulation, $F_{1,E}^{s}$ is the node $(3,5)$ in

the 6 X 6 input and template arrays. The input field has been left blank, but since there has been no template recoding in ART unit s the template displays the entire field of input stimuli. The now-active F_2^s node and the F_1^s nodes having nonzero connections in its template undergo a reverberating period of activity through both the top-down template and their corresponding, also-nonzero, bottom-up connections. The next I node to reach a below-threshold level now releases its inhibition of its F_2^s node. According to the biological interpretation of adaptive resonance theory, the F_2^s node representing $e_{2,1}$ will have had the synaptic neurotransmitter substance in its bottom-up connections depleted during its reverberating activity. As a consequence, it is now vulnerable to the competitive inhibitory input from the rejuvenated F_2^s node representing $e_{2,1}$, which is receiving the full stimulus from $F_{1,E}^s$ since the latter also has a nonzero template connection from it. The first F_2^s node is thereby suppressed and now it is the event $e_{2,2}$ that undergoes replay (Fig. 45). This process continues throughout the replay of the sequence e_2 (Figs. 46–48).

5.4 Discussion of Replay Result

The simulations presented here do not attempt to address the fine details of neural network processing; they are only intended to evaluate the ability of the supertemplate architecture to provide more discriminatory ability between sequences with different contexts during learning, as in Experiment 1, and to demonstrate that the supertemplate architecture can indeed perform the replay of a learned sequence and can be argued to have done so because it has the needed mechanisms for this. The specific mechanisms that provide the replay are (1) the activation of one or more F_1^s nodes such as $F_{1,E}^s$ that provide the stimulus that re-activates F_2^s, together with (2) the inhibition and timed event-by-event release of inhibition provided by the top-down connections from the integrator layer, which in turn is displaying the recency-gradient template currently active in ART unit s. There is one detail, however, that is worth addressing in this discussion. In the replay simulation, we assumed that there was an exact match between the F_2^s nodes having $F_{1,E}^s$ in their nonzero template connections and those receiving top-down inhibition from the current recency gradient in I. This was done to ensure that the entire F_2^s layer is initially shut down by the inhibition, for in general $F_{1,E}^s$ could have nonzero template connections with F_2^s nodes which are not part of the sequence. This assumption can be removed under the weaker assumption that the sequences which are represented separately by the architecture have unique sets of persistently-active input stimuli through F_1^s, and that the intersections of these sets are sufficiently small and the bottom-up F_1^s to F_2^s weights are also sufficiently small that only the F_2^s nodes for the events in a sequence will become fully active during its replay. Further simulations will test the applicability of the weaker assumption.

6 Conclusion

We have presented a theory of event sequence learning by a neural architecture that is based upon the categorical neural semantic theory (CNST) together with an architectural implementation of this theory that employs a system of interconnected ART and temporal integrator networks. The architecture is hierarchical, as is appropriate since the CNST expresses the incremental representation of a hierarchy of concepts and concept abstraction/specialization relationships in a diagrammatical, adaptive neural network structure. The ART and linear and nonlinear integrator neural network components, acting through their reciprocating bottom-up and top-down connections as specified, provide both the learning and replay modes that support a neural memory capable of learning the event sequences and forming appropriate generalizations of them in the context of their input stimulus components which are persistently active throughout a given sequence.

There is a great deal more work that must be done to evaluate the effectiveness and scope of this theoretical model, particularly as we propose it as a mathematical semantic model of episodic memory encoding and recall as well as motor sequence learning and other functions. There is also more theoretical work to be done; for example, we have not yet investigated in detail the transfer of information acquired during episodic encoding into semantic memory. What we do have, through bottom-up connections as indicated for ART unit s in the foregoing experiments, is the ability to build episodic representations out of semantic memory components. However,

the transfer from episodic to semantic memory is of the utmost importance, as both directions of transfer are implicated in cognitive and neural investigations of declarative memory [18]. An initial model of abstraction from episodic to semantic memory has been performed with the CNST and remains to be explored through simulation and further modeling.

Other essential investigations are the application of this work directly to motor and other sequence learning and execution by the brain, and to the purported neural processing in episodic memory storage and recall via the hippocampal formation and its interactions with other brain regions, principally through entorhinal cortex and the amygdala. However, these are challenges that we hope will be engaged in collaboration with those investigating these phenomena in cognitive and neural science.

A Graphics from the Initial Temporal Architecture Simulation

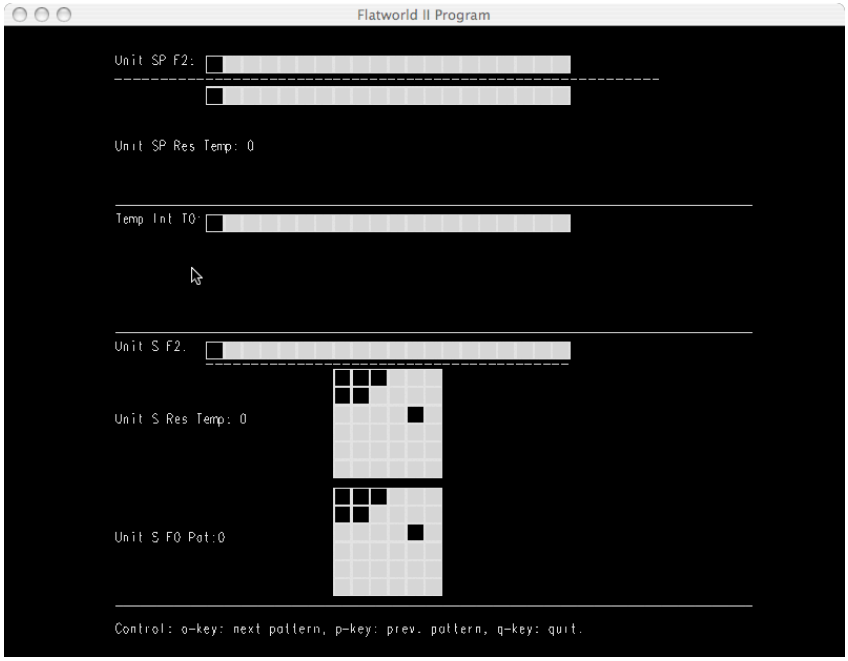


Figure 20: **Event 1 of sequence e_1.**

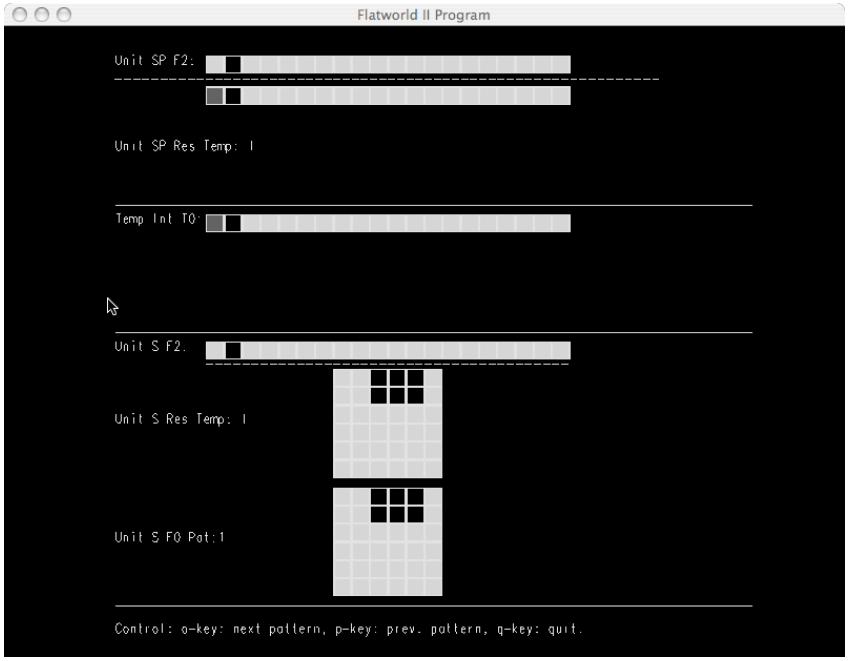


Figure 21: **Event 2 of sequence e_1.**

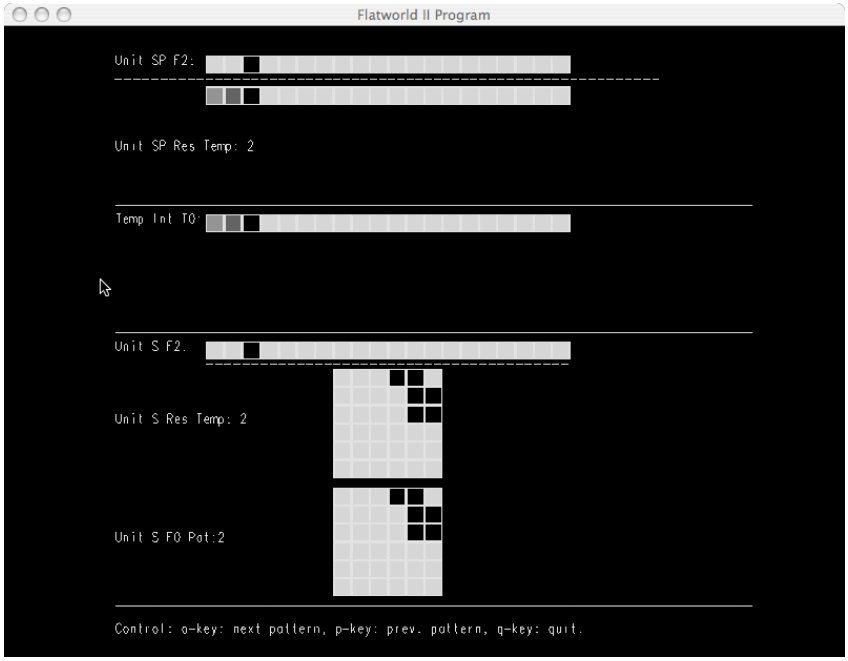


Figure 22: **Event 3 of sequence e_1.**

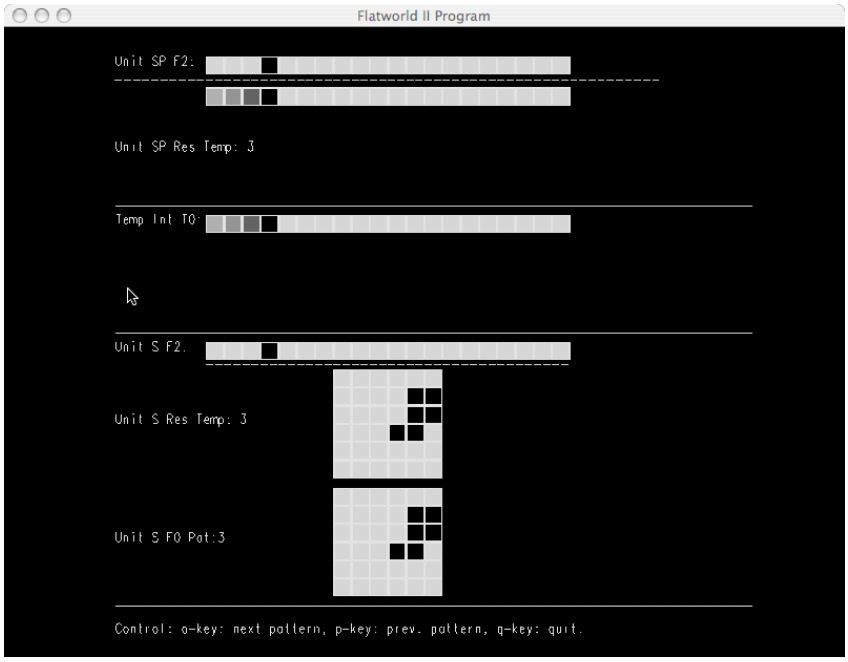


Figure 23: **Event 4 of sequence e_1.**

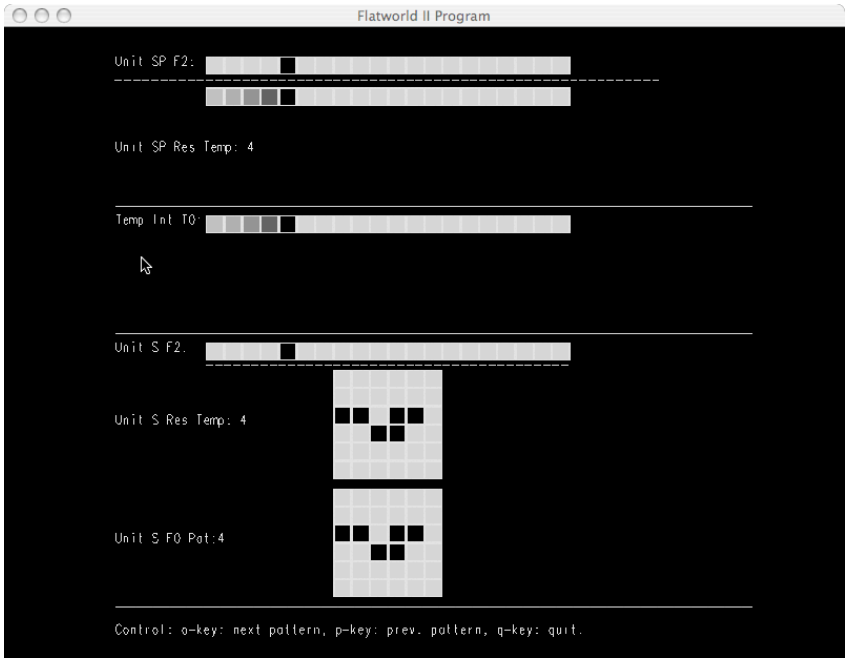


Figure 24: **Event 5 of sequence e_1. The template $Q_5^{s'}$, identical with the recency gradient at I, $F_0^{s'}$ and $F_1^{s'}$, represents the sequence at ART unit s'.**

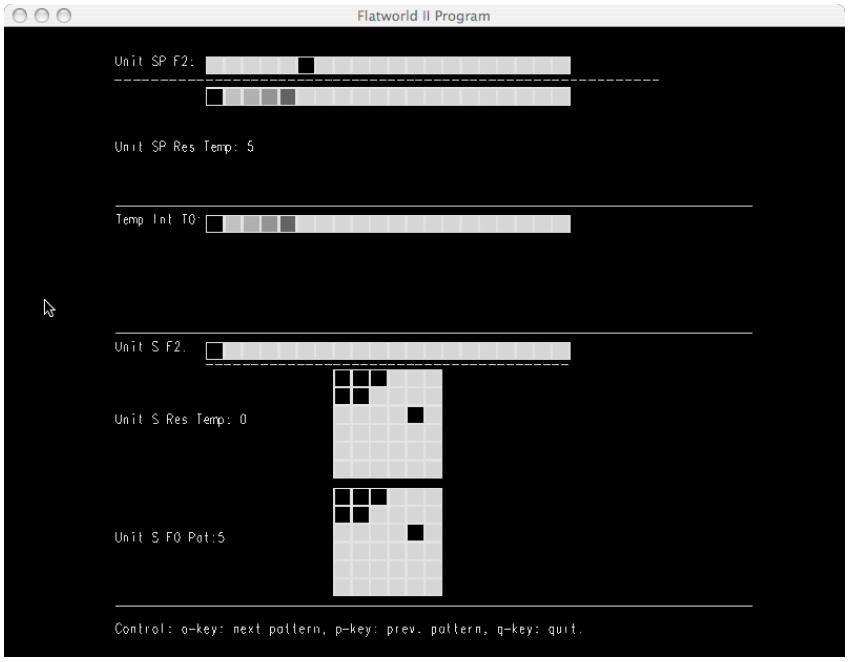


Figure 25: **Event 1 of sequence e_2 .**

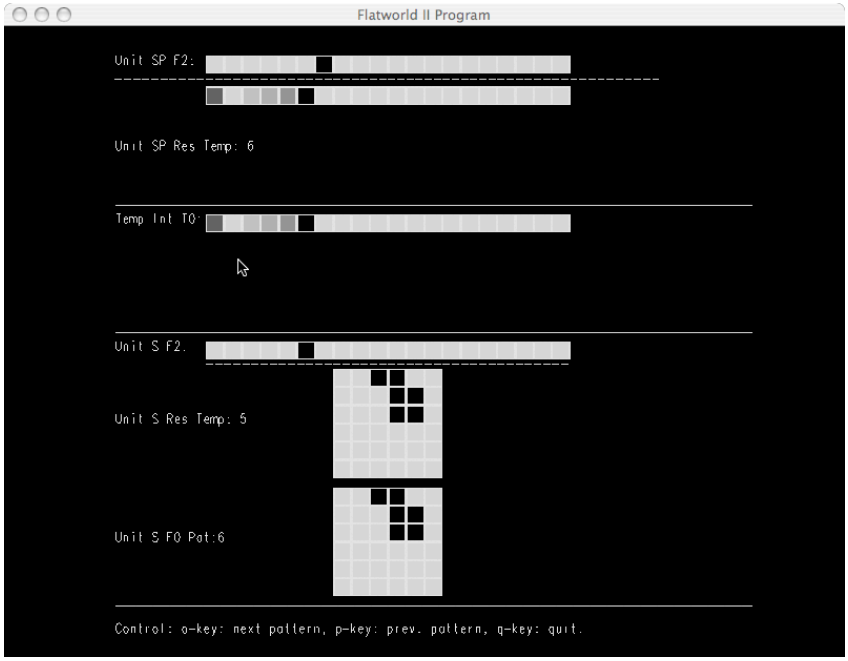


Figure 26: **Event 2 of sequence** e_2 **.**

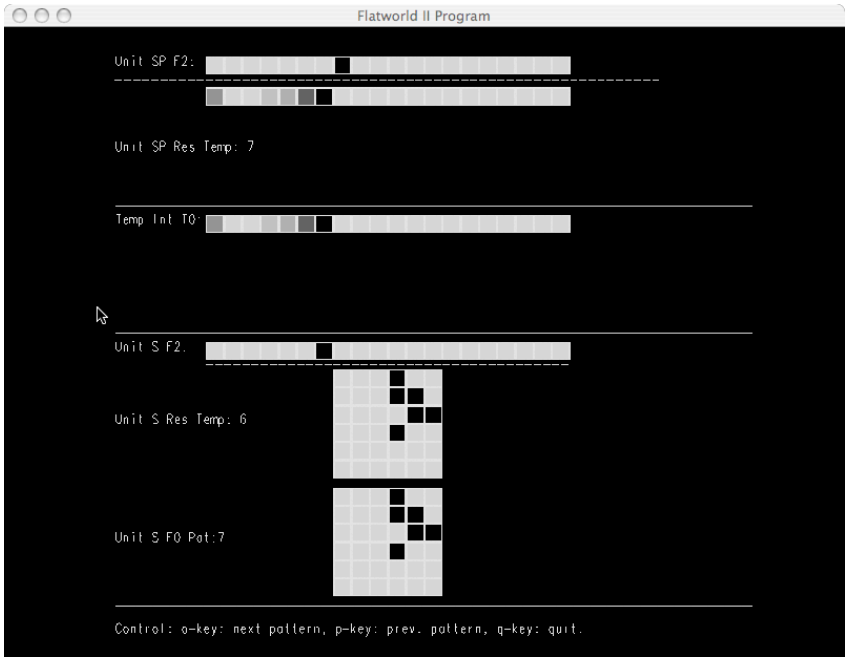


Figure 27: **Event 3 of sequence** e_2 **.**

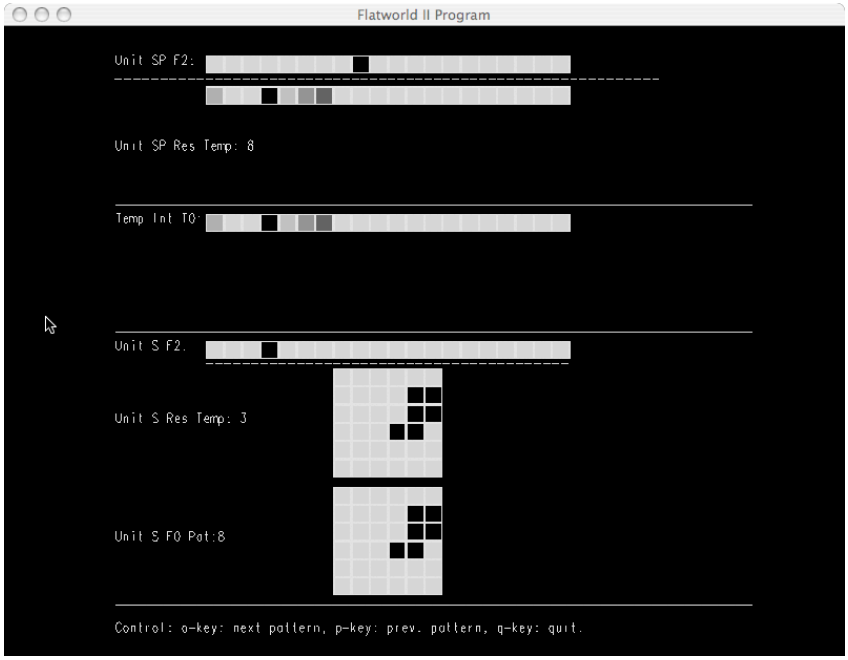


Figure 28: **Event 4 of sequence e_2 .**

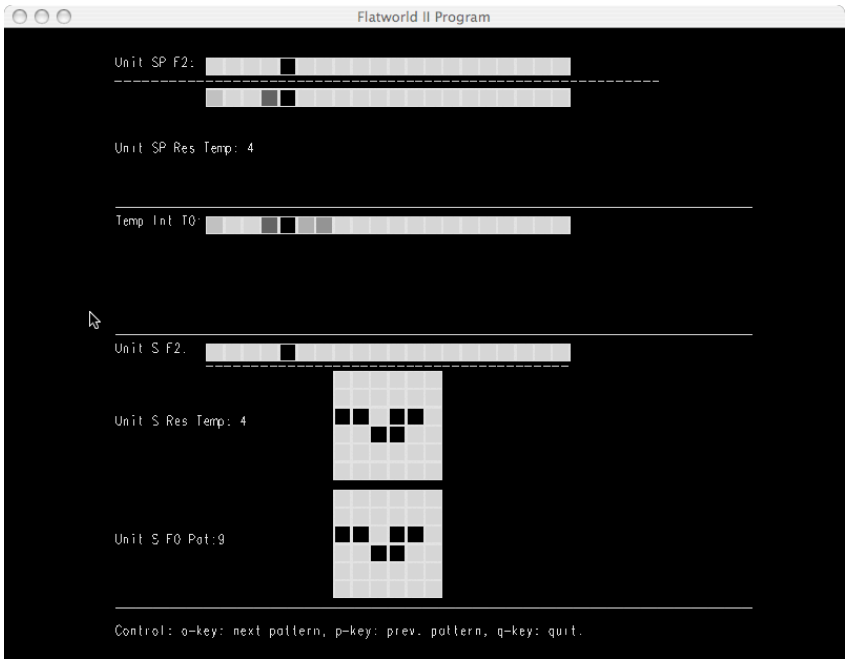


Figure 29: **Event 5 of e_2. Because of the vigilance level $\rho_{s'} = 0.75$, $Q_5^{s'}$, the template that was produced by $e_{1,5}$, is recoded by $e_{2,5}$. As a result, both sequences have the same resonant node $F_{1,5}^{s'}$. They are missing events 2 and 3 because of the mismatch there, where $\rho_s = 0.9$.**

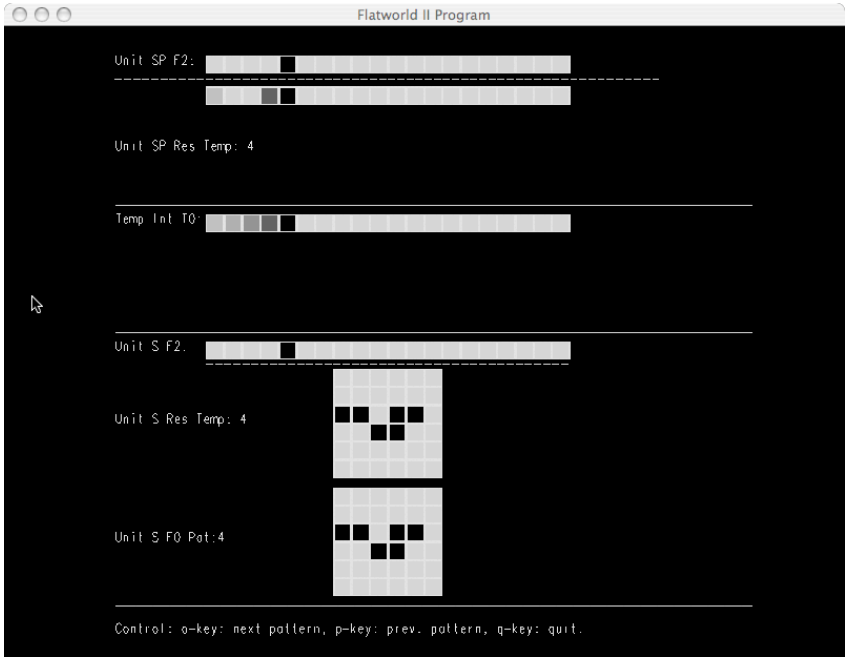


Figure 30: **At the end of the second pass through e_1, the recency gradient at $F_1^{s'}$ has the same resonant node $F_{1,5}^{s'}$, but with its template $Q_5^{s'}$ recoded by $e_{2,5}$.**

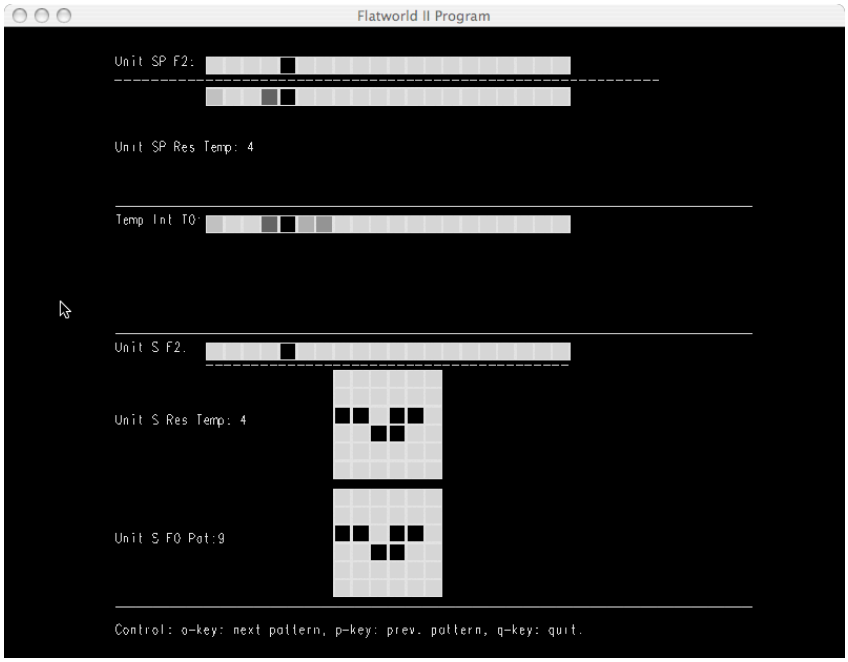


Figure 31: **At the end of the second pass through e_2, the recency gradient at $F_1^{s'}$ also retains the resonant template $Q_5^{s'}$.**

B The Supertemplate Architecture Simulation

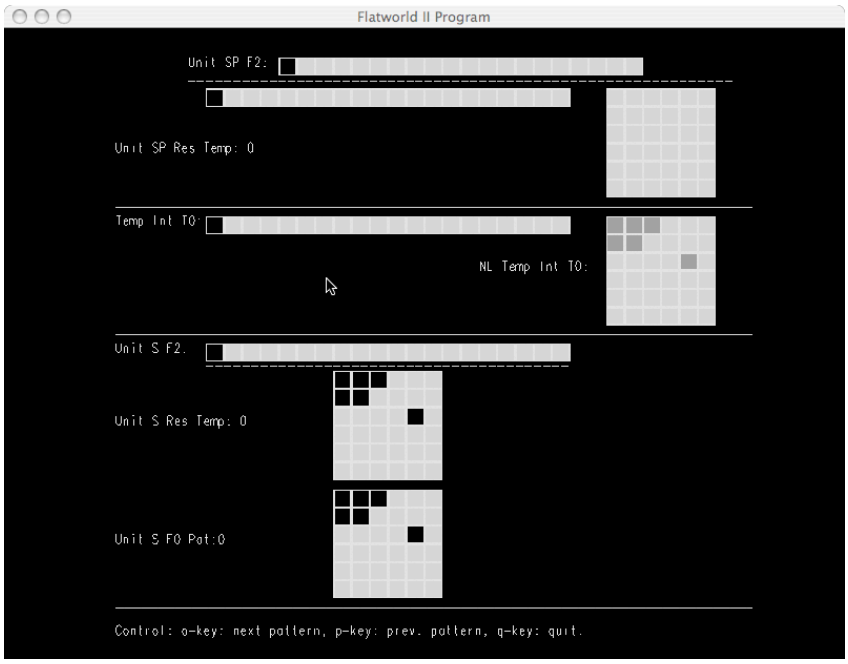


Figure 32: **With supertemplates: At $e_{1,1}$, the inputs to $F_2^{s'}$ via $F_0^{s'}$ through the weak supertemplate connections from F_1^s have not yet reached the thresholds of the $F_0^{s'}$ nodes. Hence, they are registered as zero, and the processing is the same as it is without the supertemplate connections.**

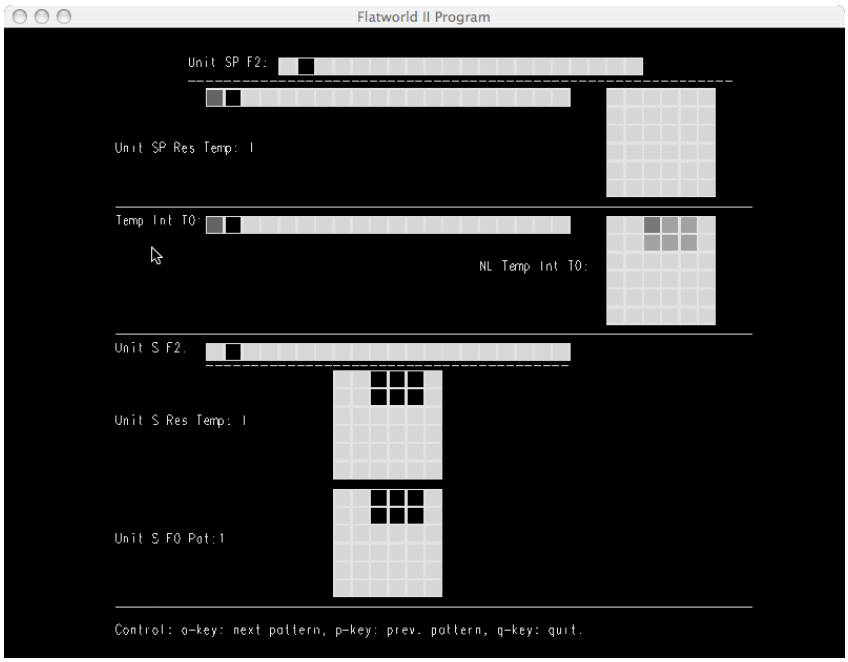


Figure 33: **At $e_{1,2}$, the inputs to $F_2^{s'}$ via $F_0^{s'}$ through the weak supertemplate connections from F_1^s are still below threshold.**

Figure 34: **At $e_{1,3}$, the processing is still the same as it is without the supertemplate connections.**

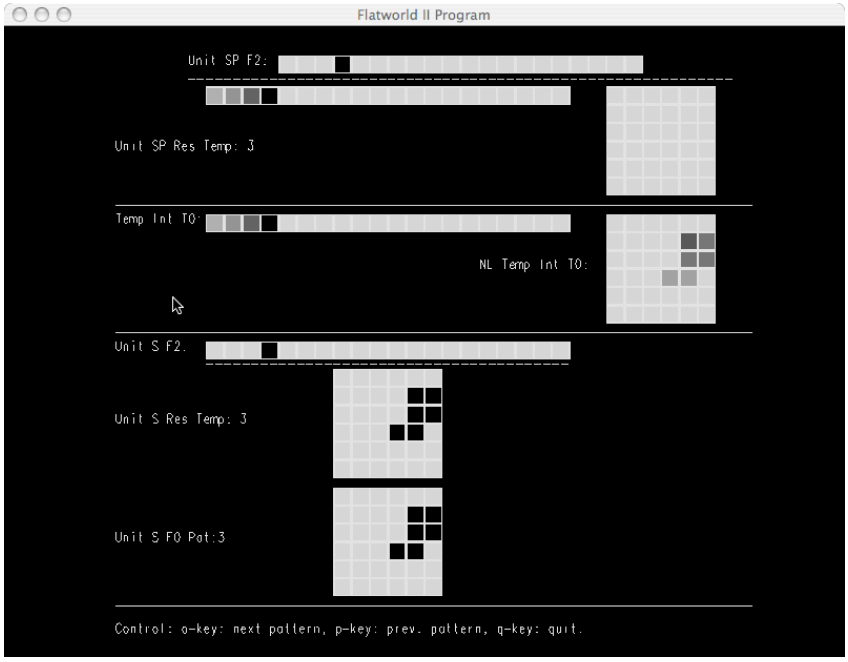


Figure 35: **At $e_{1,4}$, the processing is the same as it is without the supertemplate connections.**

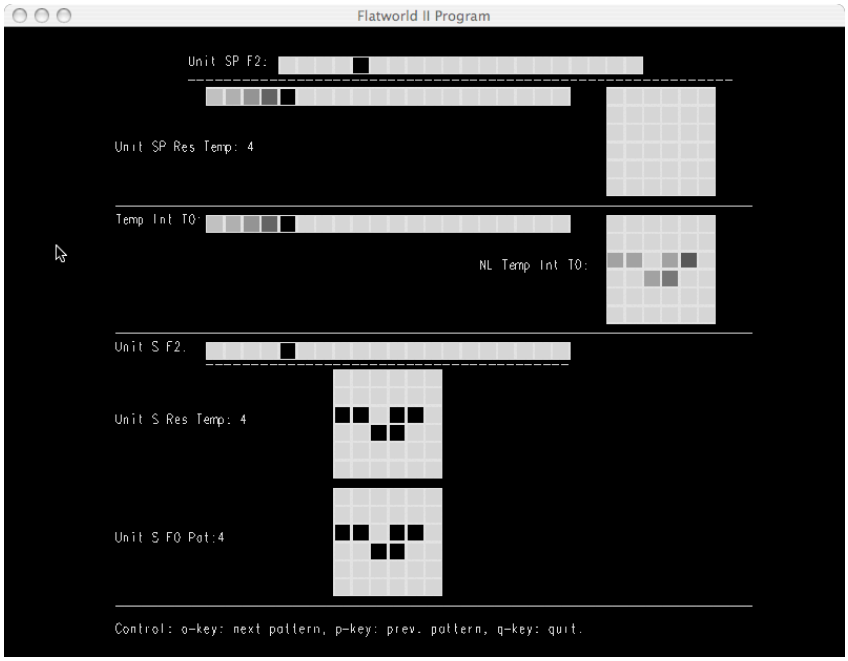


Figure 36: **Regardless of the presence of supertemplate connections, at $e_{1,5}$ the input from F_1^s remains zero because each node of F_1^s has registered zero activity during at least one of the last five events.**

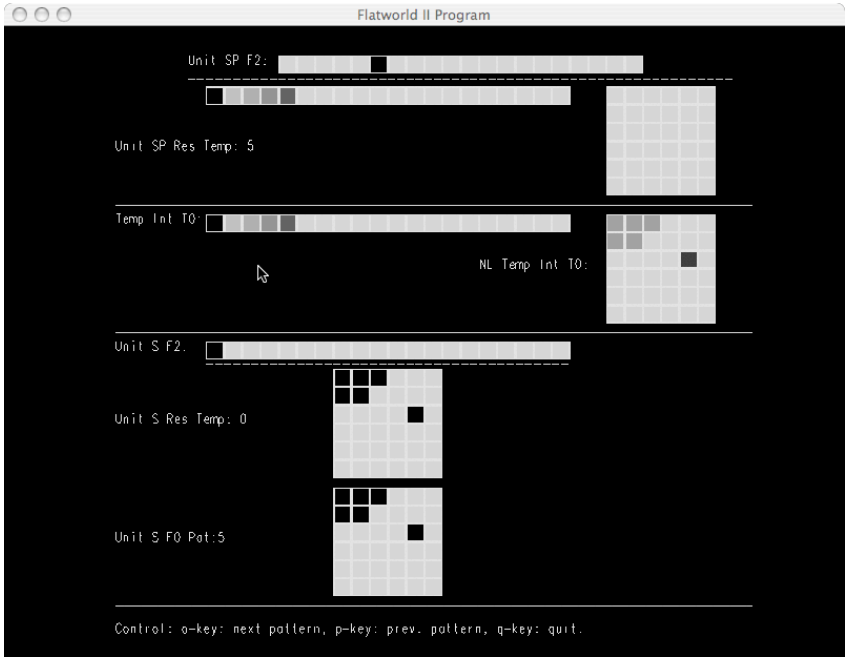


Figure 37: **At $e_{2,1}$ the same situation holds as at previous events.**

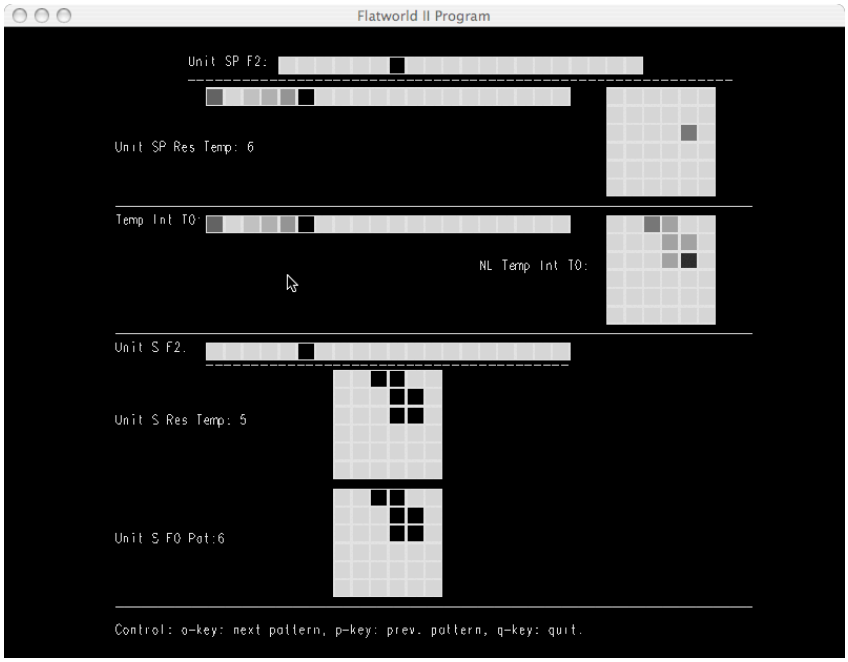


Figure 38: **Novel behavior appears during the input of $e_{2,2}$: The above-threshold input to $F_0^{s'}$ from the nonlinear integrator produces a supertemplate with a nonzero weight corresponding to the supertemplate connection for the node in row 3, column 5 of $F_1^{s'}$, which has been active in the last 5 consecutive events.**

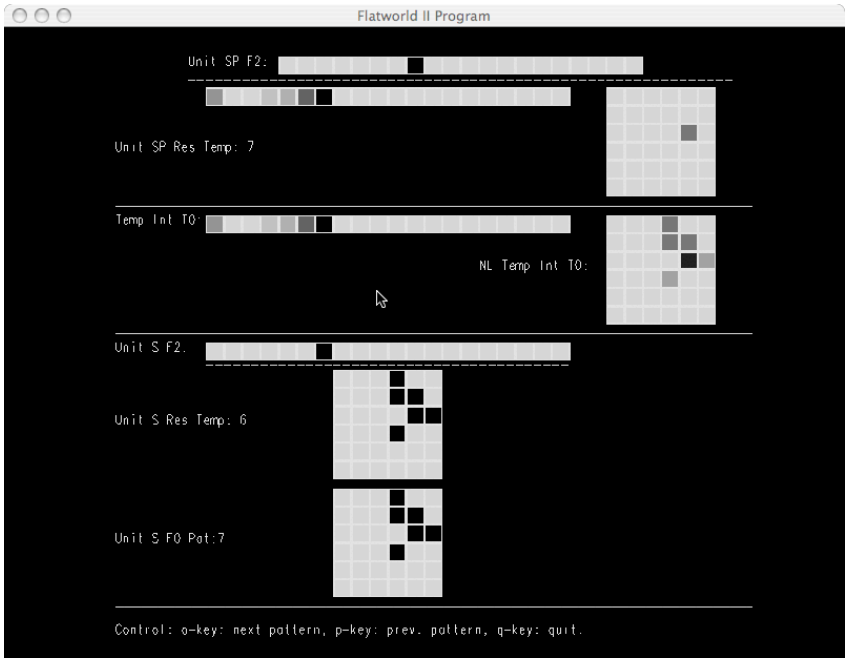


Figure 39: **At $e_{2,3}$ the supertemplate connection remains active.**

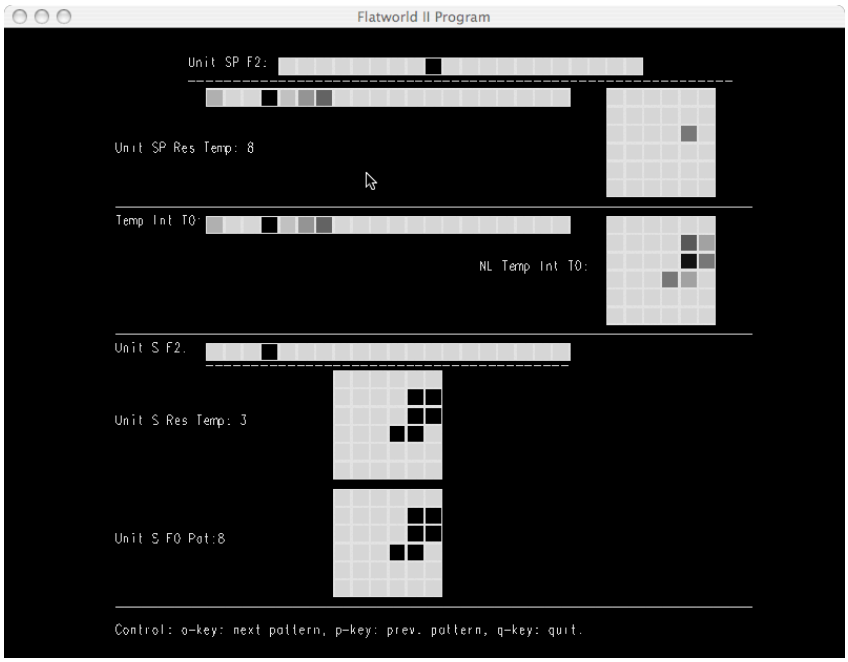


Figure 40: **At $e_{2,4}$ the supertemplate connection remains active.**

Figure 41: **Finally, at $e_{2,5}$ the supertemplate connection remains active still. With its contribution to the input pattern at $F_0^{s'}$, there is sufficient difference with the template $Q_5^{s'}$ that the latter is NOT recoded. Instead, the input of $e_{2,5}$ results in a new template, $Q_{10}^{s'}$. Notice the above-threshold input to $F_0^{s'}$ through the supertemplate connection indicated by row 3, column 5 of $F_0^{s'}$, which has held since the event $e_{1,3}$. We now have separate templates for e_1 and e_2, expressing not only their recency gradients but the blending object that is consistent throughout e_2.**

Figure 42: **Event 5 of sequence 1, second pass. Notice that supertemplate $Q_5^{s'}$ is recalled, unaltered by the processing of sequence e_2.**

Figure 43: **Event 5 of sequence 2, second pass. Supertemplate $Q_{10}^{s'}$ is recalled.**

C Sequence e_2 replay

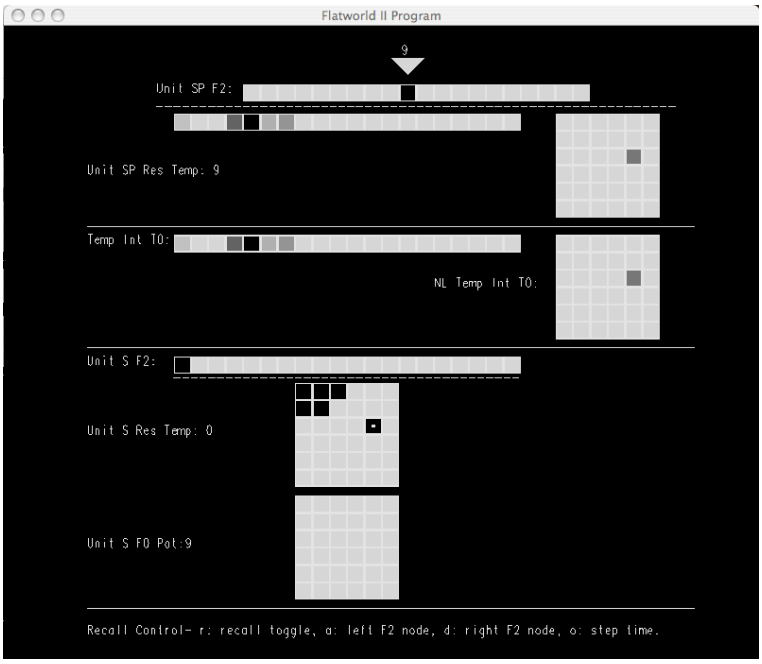


Figure 44: **Replay of event $e_{2,1}$, time step 1 encoded via supertemplate $Q^{s'}_{10}$. The small white square identifies F^s_1 node $(3,5)$, the single persistent node for sequence e_2.**

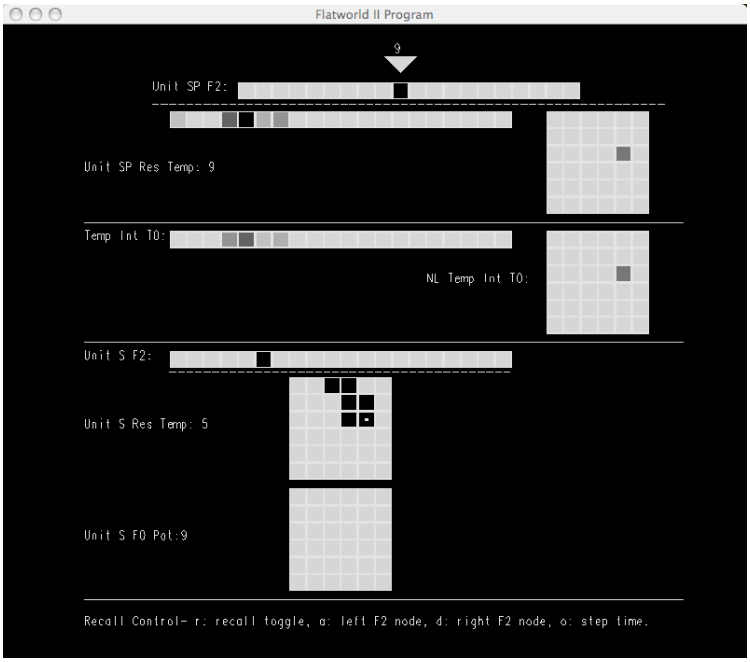


Figure 45: **Replay of event $e_{2,2}$ encoded via supertemplate $Q^{s'}_{10}$.**

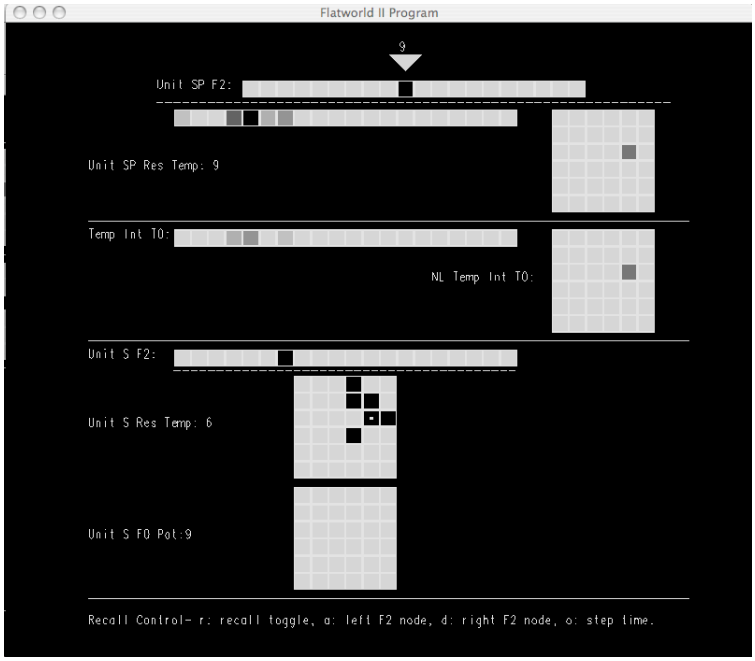


Figure 46: **Replay of event $e_{2,3}$ encoded via supertemplate $Q_{10}^{s'}$.**

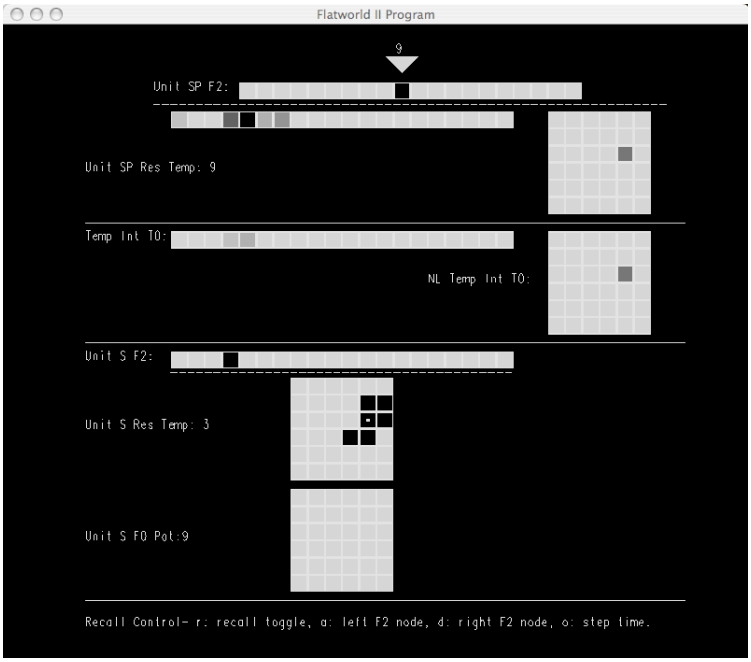


Figure 47: **Replay of event $e_{2,4}$ encoded via supertemplate $Q_{10}^{s'}$.**

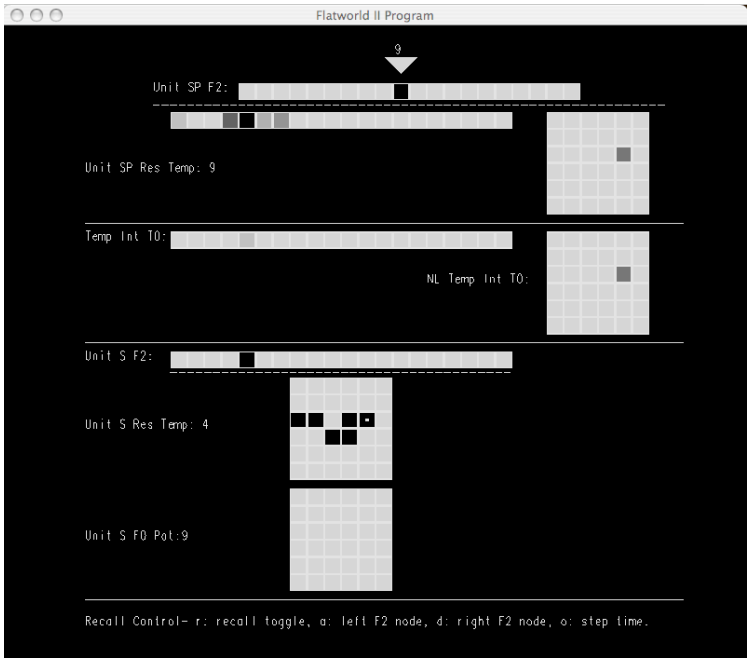


Figure 48: **Replay of event $e_{2,5}$ encoded via supertemplate $Q_{10}^{s'}$.**

References

[1] Hisham E. Atallah, Michael J. Frank, and Randall C. OReilly. Hippocampus, cortex, and basal ganglia: Insights from computational models of complementary learning systems. *Neurobiology of Learning and Memory*, 82:253–267, 2004.

[2] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[3] Roy L. Crole. *Categories for Types*. Cambridge University Press, 1993.

[4] A. C. Ehresmann and J.-P. Vanbremeersch. Information processing and symmetry-breaking in memory evolutive systems. *BioSystems*, 43:25–40, 1997.

[5] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[6] H. Gust and K.-U. Kühnberger. Learning symbolic inferences with neural networks. In *Proceedings of the XXVII Annual Conference of the Cognitive Science Society (CogSci2005)*. Cognitive Science Society, 2005.

[7] M. J. Healy, R. D. Olinger, R. J. Young, S. E. Taylor, T. P. Caudell, and K. W. Larson. Applying category theory to improve the performance of a neural architecture. *Neurocomputing*, 72:3158–3173, 2009.

[8] Michael J. Healy and Thomas P. Caudell. From categorical semantics to neural network design. In *The Proceedings of the IJCNN 2003 International Joint Conference on Neural Networks. Portland, OR, USA. (CD-ROM proceedings)*, pages 1981–1986. IEEE,INNS, IEEE Press, 2003.

[9] Michael J. Healy and Thomas P. Caudell. Generalized lattices express parallel distributed concept learning. In Vassilis G. Kaburlasos and Gerhard X. Ritter, editors, *Computational Intelligence Based on Lattice Theory*, volume 67 of *Studies in Computational Intelligence*, pages 59–77. Springer, 2007.

[10] Michael John Healy and Thomas Preston Caudell. Ontologies and worlds in category theory: Implications for neural systems. *Axiomathes*, 16(1-2):165–214, 2006.

[11] Michael John Healy, Thomas Preston Caudell, and Timothy E. Goldsmith. A model of human categorization and similarity based upon category theory. Technical Report EECE-TR-08-0010, Department of Electrical and Computer Engineering, University of New Mexico, June 2008.

[12] Ole Jensen and John E. Lisman. Hippocampal sequence-encoding driven by a cortical multi-item working memory buffer. *TRENDS in Neurosciences*, 28(2):67–72, 2005.

[13] F. W. Lawvere and S. H. Schanuel. *Conceptual Mathematics: A First Introduction to Categories*. Cambridge University Press, 1995.

[14] P. A. Lipton and H. Eichenbaum. Complementary roles of hippocampus and medial entorhinal cortex in episodic memory. *Neural Plasticity*, 2008:1–8, 2008.

[15] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[16] J. Meseguer. General logics. In H.-D. Ebbinghaus *et al*, editor, *Logic Colloquium '87*, pages 275–329. Science Publishers B. V. (North-Holland), 1989.

[17] B. C. Pierce. *Basic Category Theory for Computer Scientists*. MIT Press, 1991.

[18] Endel Tulving. *Elements of Episodic Memory*. Oxford University Press, New York, 1983.

[19] Endel Tulving. What is episodic memory? *Current Directions in Psychological Science*, 2:67–70, 1993.

[20] Robert J. Young, Mike Ritthaler, Peter Zimmer, John McGraw, Michael J. Healy, and Thomas P. Caudell. Comparison of adaptive resonance theory neural networks for astronomical region of interest detection and noise characterization. In *2007 IEEE-INNS International Joint Conference on Neural Networks. Orlando, FL. (CD-ROM proceedings)*, page No. 1541. IEEE,INNS, IEEE Press, 2007.