Advances in Cognitive Systems 8 (2019) 113–131 Submitted 5/2019; published 12/2019

(Sub)Symbolic × (A)Symmetric × (Non)Combinatory: A Map of AI Approaches Spanning Symbolic/Statistical to Neural/ML

Paul S. Rosenbloom ROSENBLOOM@USC.EDU
Department of Computer Science and Institute for Creative Technologies, University of Southern California, 12015 Waterfront Drive, Playa Vista, CA 90094 USA

Himanshu Joshi HIMANSHU@QUALCOMM.COM
Qualcomm, 6775 Morehouse Drive, San Diego, CA 92121 USA

Volkan Ustun USTUN@ICT.USC.EDU
Institute for Creative Technologies, University of Southern California, 12015 Waterfront Drive, Playa Vista, CA 90094 USA

Abstract

The traditional symbolic versus subsymbolic dichotomy can be decomposed into three more basic dichotomies, to yield a 3D (2×2×2) space in which symbolic/statistical and neural/ML approaches to intelligence appear in opposite corners. Filling in all eight resulting cells then yields a map that spans a number of standard AI approaches plus a few that may be less familiar. Based on this map, four hypotheses are articulated, explored, and evaluated concerning its relevance to both a deeper understanding of the field of AI as a whole and the general capabilities required in complete AI/cognitive systems.

1. Introduction

Science, in its broadest form, concerns understanding the "world," itself in its broadest form. Meta-science then is a branch of science that focuses on understanding science as part of the world. Rosenbloom (2012), for example, showed how the computing sciences could be understood in terms of how they relate to both themselves and the physical, life, and social sciences. In particular, a pair of primitive relationships – implementation and interaction – were together shown to yield a map of the computing sciences and its relationship to the other three domains. The relationship between symbolic and neural approaches to artificial intelligence (AI) is also often characterized via one of these two relationships; that is, either as a neural implementation of symbolic processing (e.g., Cho, Rosenbloom, & Dolan, 1991; Smolensky & Legendre, 2006) or as separate neural and symbolic modules that interact (e.g., Jilk et al., 2008; Sun, 2016). Alternatively, the two may simply be characterized as polar opposites on a grand symbolic-versus-subsymbolic dichotomy.

From a symbolic, cognitive-systems perspective, understanding this relationship more deeply is more critical now than ever before, as the dramatic successes of the modern neural resurgence – in the form of what is now often called deep neural networks or deep learning – is leading it to an increasing domination of the overall conversation. The initial goal of this article is to further such an understanding by decomposing the traditional, but as it turns out rather coarse-grained, symbolic-versus-subsymbolic dichotomy into three finer-grained ones. Using the title notation, these are: (1) *(non)symbolic*; (2) *(a)symmetric*; and (3) *(non)combinatory*. Each of these three, whether seemingly familiar or not, is given a particular definition later that yields an orthogonal dimension along which it turns out that numerous AI approaches can be characterized.

These finer-grained dichotomies are first applied to understanding what will be referred to as symbolic/statistical versus neural/ML processing. The former term acknowledges that traditional symbolic processing can be considered as a specialization of the broader notion of statistical relational processing (Getoor & Taskar, 2007), where the probabilities are limited to 0 and 1. The recently proposed Common Model of Cognition (Laird, Lebiere & Rosenbloom, 2017) further argues that cognitive architectures – i.e., models of the fixed structures and processes that implement a mind (Langley, Laird, & Rogers, 2009; Kotseruba & Tsotsos, 2018) – require not only symbols but also quantitative metadata, such as probabilities and utilities. The latter term recognizes that the recent AI boom is due primarily to large-scale applications of learning in neural networks. Use of the term neural/ML is not intended to imply that all learning is necessarily neural in form, but it is intended to refer to the vast range of work in machine learning that currently is.

Based on the 3D space engendered by the cross product of these finer-grained dichotomies, four hypotheses are then articulated, explored, and evaluated, although while varying considerably in how speculative they are and in how well they are evaluated. Together, the four hypotheses end up not only bearing on a better understanding of the original grand dichotomy, but also on the computational approaches that make up the field of AI as a whole and those approaches that must be part of complete cognitive systems. The first two hypotheses focus on AI as a whole and provide the core of the contribution here, whereas the latter two hypotheses raise more speculative possibilities for analyzing and understanding particular cognitive systems. Only very preliminary evaluations are provided for the latter two, but the intent at this point with respect to them is mainly to make the case that they are worth further exploration.

The first core hypothesis addresses the initial goal of better understanding the original grand dichotomy, by claiming that archetypical symbolic/statistical processing is characterized by the set of first terms in each of the three finer-grained ones and archetypical neural/ML processing by the set of second terms. This may appear to make sense only for the first and third – which are discussed further in Sections 2 and 4, respectively – but the claim is that it is also true of the second, which will be explained and discussed further in Section 3. Each dichotomy is analyzed in terms of a proposal for a central integrating idea that aids in both understanding and spanning it, plus additional sub-dichotomies that help to dissect the three main ones much as they themselves help dissect the single traditional dichotomy. This hypothesis is by necessity evaluated informally, according to whether by the end of Section 4 the arguments for these dichotomies are found to be compelling.

Given the first core hypothesis, symbolic/statistical and neural/ML approaches can be situated at opposite corners of a 3D space with eight (2×2×2) total cells, but with a whole structured space of intermediate points that share one or more choices with each of them (Table 1). The second core hypothesis is then that this space frames a novel form of map over a range of approaches that can help understand both the overall topology of AI and the core commonalities and differences among

Table 1. 3D map based on the cross product of the dichotomies, with separate planes for combinatory versus noncombinatory.

Combinatory	*Symbolic*	*Subsymbolic*
Symmetric	Symbolic/Statistical	
Asymmetric		

Noncombinatory	*Symbolic*	*Subsymbolic*
Symmetric		
Asymmetric		Neural/ML

the approaches that populate the map (Section 5). It is important to note, though, that this hypothesis does not claim that every AI approach fits within some cell of this map. Some approaches may require combinations of cells, whereas others may fall completely outside of the map. Both possibilities are worth further investigation, and a bit more will be said about them later, but the focus here is on the map's spanning a substantial enough swath of approaches within individual cells.

This hypothesis is evaluated by fleshing out all eight cells with standard AI approaches whenever possible and with less well-known ones as necessary. This is not an exhaustive analysis of all AI approaches – investigating such a possibility will be left for future work – but each cell will ultimately contain one or more AI approaches that are distinct from those in the other cells. Furthermore, the structure of the map will highlight a number of key commonalities and differences among the approaches.

Several prior maps have been proposed that bear a family resemblance to the one here, at least three of which include a variant of (a)symmetry (Anderson & Lebiere, 1998; Sun, 2016; Rosenbloom, 2019), although the first two under other names – procedural versus declarative and action-centered versus non-action-centered – with the third including a discussion of how the other two can be viewed as variants of (a)symmetry. The first two also include variations on (sub)symbolic, one under this name and the other as explicit versus implicit. The first of these prior maps also includes a third dimension, for performance versus learning, but the focus in this article is almost exclusively on performance, with the task of extending the map to learning left to future work.

One additional source of agreement among these three earlier maps is that they are to be used in analyzing particular cognitive architectures – ACT-R, CLARION, and Sigma, respectively – rather than for mapping the space of AI approaches. In this way they also bear a relationship to prior taxonomic and componential analyses of cognitive architectures, such as those found in Kotseruba and Tsotsos (2018). However, both of these latter forms of structures are weaker predictively than maps. Taxonomies do not include more choices than are needed for the initial set of ideas and thus do not naturally generate empty regions where unknown possibilities may be found. Componential analyses do not naturally provide the systematic structuring that yields insights into commonalities and differences among the elements.

The core purpose here shares with these prior efforts the goal of developing a map rather than either a taxonomy or a componential analysis, but it differs in focusing on the structure of the field of AI rather than on the structure of individual cognitive architectures. At least one prior map

(Minsky, Singh, & Sloman, 2004) does share this core purpose, but its dimensions concern properties of tasks that are hypothesized to distinguish subspaces over which different AI approaches are most effective, rather than aspects of how the approaches themselves function. Whether such different forms of maps over the space of AI approaches can ultimately be related in a useful manner poses an interesting question for future consideration.

The other purpose, of understanding individual cognitive architectures, does turn out to still play a role in this article in inspiring the two additional hypotheses. The first such hypothesis is that every complete cognitive system – i.e., one with full human-level intelligence – must support some approach in each cell of Table 1. This follows the general path of the previous work, although establishing the necessity of such a map can itself be quite involved. For now, we will simply note that two of the three dichotomies here align approximately with two of the previous efforts, with only (non)combinatory being new as a distinct dichotomy.

In Section 6, this additional hypothesis is explored with respect to the Sigma cognitive system (Rosenbloom, Demski, & Ustun, 2016a), which is of particular interest in this context because of the broad range of AI approaches it can embody supra-architecturally (Rosenbloom, 2015) – that is, based on the architecture plus knowledge and skills encoded within it – that are capable of operating in parallel without either interfering with each other above the architecture or requiring a separate module for each within the architecture. This analysis yields a map that is similar in kind to the other three just mentioned, but its focus goes beyond just what is natively provided by the architecture to also include key supra-architectural capabilities that are part of the larger cognitive system it supports.

The second additional hypothesis goes beyond the first by proposing that every complete cognitive system must support not just some approach in each cell, but an approach within each cell that captures the essence of what makes it distinctly useful, and has thus justified individual researchers or whole communities focusing on it in isolation. For example, although linear, subsymbolic, activation-passing networks sit in the bottom-right cell of the table, they achieve nothing like the power provided by deep learning, and thus would not be considered to have captured the essence of what makes this cell useful. Capturing these essences across the whole table is beyond the scope of this paper; however, as with the previous hypothesis, a preliminary application to Sigma will be explored, at the end of Section 6, by analyzing which of the AI approaches in the map are, or potentially could be, available to it. Section 7 then summarizes and concludes.

2. Symbolic versus Subsymbolic

A classical definition of symbols concerns entities that can be combined into expressions (Newell & Simon, 1976). Notions of designation and interpretation are also then included that enable expressions to represent, and for what is represented to be viewed as a program to be executed. Classical subsymbolic representations instead center on entities, i.e., units, that neither designate nor represent but do have associated numbers. In distributed formulations, it is typically specific vectors of associated numbers that represent (e.g., Pennington, Socher, & Manning, 2014).

The goal driving the development of the Common Model of Cognition is to achieve a community consensus concerning the structures and processes that jointly yield human-like minds. As part of the current consensus, the Common Model provides the proposed central integrating concept here of *dyads of symbolic data and quantitative metadata: <d, m>*. However, the notion of symbol in the Common Model is weaker than Newell and Simon's, including only that symbols are primitive elements that can be combined. Here, we weaken this even further, to a primitive element to which quantitative metadata can be attached (with the combinatory aspect separated out into its own dichotomy). The metadata may consist of probabilities or activations, as are of concern in Section 3, or other kinds of numeric annotations.

Given the idea of dyads, there are three sub-dichotomies that help further understand the (sub)symbolic dichotomy. The first concerns how much of a dyad's meaning is determined by the symbol/data versus the number/metadata. In a pure symbol system, all of the meaning is in the symbol. The number, should it be present, is just 0 or 1, for false or true in a probabilistic interpretation. In contrast, in a pure neural system, the symbols would simply be placeholders identifying positions within a vector, with the meaning all conveyed by the numbers at their particular positions in the vector.

There are however, intermediate points along this sub-dichotomy – making it in reality more of a metric dimension – such as probabilistic graphical models that combine meaningful symbols with probabilities (Koller & Friedman, 2009), local connectionist networks that combine meaningful symbols with activations (McClelland & Rumelhart, 1981), and one-hot encodings that have non-zero metadata at only one location (Harris & Harris, 2012). Whether a dyad should be considered symbolic versus subsymbolic thus effectively comes down to how much of its meaning is in its data versus its metadata. Still, the entire dyad is always assumed to exist, even if/when some parts are implicit.

The second sub-dichotomy concerns whether there is nonlinearity in the processing of the data versus the metadata. Data nonlinearity in general consists of the classical discrete changes at the heart of symbol processing, such as replacing a symbol in an expression with another, combining multiple expressions, or firing rules that generate new expressions. Metadata nonlinearity typically involves continuous, or even differentiable, nonlinear changes in numerical values; for example, via a sigmoid, RELU, tanh, or softmax. In one form, data nonlinearity can be viewed as step functions – from 0 to 1 or vice versa – in metadata; however, it can also go beyond this to the creation and deletion of elements to which metadata can be attached.

This sub-dichotomy is closely related to the first, as the locus of nonlinearity tends to follow the locus of meaning. But whether there is nonlinearity in the data or the metadata, or both, full intelligence appears to require it somewhere – implying a potentially interesting hypothesis in its own right, but not one that will be explored further here. Standard symbol processing is, in particular, impossible without data nonlinearity. Likewise, Choi and Darwiche (2018) discuss how metadata nonlinearity enables neural networks to significantly outperform Bayesian networks by increasing their expressibility for function fitting.

There is an intriguing body of recent work that seeks to replace standard forms of data nonlinearity with (differentiable) metadata nonlinearity, as for example in the form of Neural Turing Machines (Graves, Wayne, & Danihelka, 2014). Yet, even when this is done, critical additional forms of data nonlinearity turn up, whether for experience replay (Lin, 1992), search over state spaces (Silver et al., 2018), or other uses. Given this, and the background assumptions here, it remains critical to continue direct explorations of both forms of nonlinearity.

The third sub-dichotomy concerns whether the meaning of the data is defined via declarative versus procedural semantics. In logic, semantics provides meaning for syntactic elements in terms of the possible worlds in which the elements are true. The notion of declarative semantics follows closely on this but can be thought of more broadly in terms of a fixed, a priori meaning specification that is unaffected by how the elements are used. With procedural semantics, in contrast, the meaning of syntactic elements is an implicit consequence of how they are used.

As with logics, symbolic systems prototypically embody declarative semantics. Subsymbolic systems prototypically embody procedural semantics, as when hidden units in a neural network are ascribed meaning only through a post-learning analysis. However, rules, which are typically considered symbolic, often have procedural semantics – at least in how they are used as productions in cognitive systems rather than in their logical usage as implications or Horn clauses – where the meaning of an element is principally determined by how it is (re)used across the system. Likewise, pixels, which are typically considered subsymbolic, do have a fixed meaning, in terms of the properties of light in a particular location.

Taking these three sub-dichotomies together, symbolic systems canonically focus on meaning as fixed interpretations of (nonlinearly transformed) data, whereas subsymbolic systems focus on emergent interpretations of (nonlinearly transformed) metadata. However, given how these sub-dichotomies correlate, it can still make sense to view symbolic systems as simply deriving their meaning largely from data and subsymbolic systems from metadata.

A particularly illustrative example here is Markov networks versus Boltzmann machines (Ackley et al., 1985). The former are forms of probabilistic graphical models, like Bayesian networks, but they use symmetric rather than asymmetric links between nodes and define functions over cliques of nodes. The latter are symmetric neural networks – in contrast to the asymmetry of feed-forward neural networks – that are closely related to Hopfield networks (Hopfield, 1982). Mechanistically, Markov networks and Boltzmann machines are indistinguishable (Jordan & Sejnowski, 2001), but the former is viewed as symbolic because it has declarative data semantics whereas the latter is viewed as subsymbolic because it has procedural metadata semantics. A second example concerns rules, which as just mentioned have procedural data semantics but discrete data changes and (typically) no numbers, and so are viewed as symbolic even without declarative semantics.

3. Symmetric versus Asymmetric

Variants of the (a)symmetric dichotomy – such as shallow versus deep [reasoning] (Hart, 1982), heteroassociative versus autoassociative [networks] (Rizzuto & Kahana, 2001), declarative versus procedural [memories], action-centered versus non-action-centered [subsystems] (Sun, 2016), and model-based versus function-based [approaches] (Choi & Darwiche, 2018) – have been around for many years, although it has only recently been articulated in this form, with these earlier variants all then being mapped onto it (Rosenbloom, 2019). It is this form then that enables the central integrating concept. Consider a graph with nodes, plus links along which dyads may flow. The *directionality of dyad flow* along such links – and in particular whether in both directions versus only one – yields the proposed central integrating concept.

Logics and probabilistic graphical models provide canonical symmetric examples, with inference conceived of as dyad flow. In particular, the semantics behind standard logics permits deduction in any direction over logical sentences and the semantics of probabilistic graphical models requires the symmetric flow of messages, as in the sum-product algorithm (Kschischang, Frey, & Loeliger, 2001), to yield correct marginals or modes over the included random variables. Even when there are asymmetric notations included – such as for implications in logic and directed arcs in Bayesian networks (Pearl, 1988) – they denote an asymmetry in the semantics that is not reflected in inference direction. Implications can be used deductively in both directions and can in fact be converted into equivalent symmetric notations. Likewise, sum-product message passing is symmetric across the directed arcs in Bayesian networks.

In contrast is the asymmetric/unidirectional flow of dyads in rules and feedforward neural networks. Backwards flow is possible in such structures, in support for example of abduction, planning, or learning, but the meaning of the reverse flow is quite different – in logical terms, it is not deductive. For example, backward flows in rules are abductive and in neural networks they represent errors, instead of standard dyads, that support induction via backpropagation. Such backward flows may be considered as additional forward flows, but of a different sort along distinct asymmetric paths. Fitting this pattern are both backpropagation (Rosenbloom, Demski, & Ustun, 2017) and abduction in Sigma, based on unidirectional arcs of forward-backward processing, and bidirectional RNNs (Schuster & Paliwal, 1997).

Similar conclusions arise from contrasting learning in probabilistic graphical models versus feedforward neural networks. In the former, learning can occur locally at nodes in the network based on messages arriving in all directions along its symmetric arcs (Russell et al., 1995; Rosenbloom et al., 2013). Normal bidirectional processing in graphical models maintains a single semantics for messages in both directions along links – representing distributions over the variables on the link – while thus enabling message computations at nodes to occur in ignorance of their direction. Processing in both directions is essentially deductive.

In neural networks, learning can also occur locally at nodes in the network. This is typically conceived of as sending distinct messages – based on errors – backwards along the same links, but in a manner that enables them to be distinguished from the forward messages. However, it can also be conceived of as interacting flows forward along one path and backwards along another (Rosenbloom, Demski, & Ustun, 2017). In this latter formulation, the nodes are distinct along the two paths, but the backward nodes at which learning occurs are *tied* to the forward nodes so that what is learned from the backward path impacts the weights in the forward path. Both probabilistic and neural learning can thus occur locally via gradient descent at nodes in the network.

A related sub-dichotomy concerns probabilistic versus activational metadata. Probabilities are fundamentally likelihoods that an event will occur or that a fact will be true. When summed over all possibilities, the total likelihood equals one, although unnormalized, non-negative distributions may also be considered as "potential" probability distributions. Activations originated as numeric levels on neuron outputs, although the term is also now used more broadly in cognitive science to denote levels of interest or importance. Depending on how they are conceived, activations may range over real numbers or just non-negative or positive numbers. Despite that probabilities and activations can both yield forms of quantitative metadata, and that normalization or softmax can convert the latter to the former, much remains different between the probabilities found in standard AI systems and the activations typically found in neural and other cognitive approaches.

Table 2. 2D tensor, or binary tuple, with a single joint distribution over the instances.

Color\Shape	*Square*	*Circle*
Red	.3	.4
Blue	.2	.1

The relationship of this sub-dichotomy to (a)symmetry stems from the need for symmetry in full probabilistic reasoning, such as is found in probabilistic graphical models, whereas activation processing is typically asymmetric. There are, however, recent examples of asymmetric processing of probabilities, such as via arithmetic circuits (Darwiche, 2002) and sum-product networks (Poon & Domingos, 2011), which can provide particularly efficient probabilistic computations for a restricted subset of the entire space of problems specifiable via probabilistic graphical models, much as how description logics (Baader et al., 2003) relate to full first-order logic. There are also symmetrical neural approaches – such as Boltzmann machines – although, given their mapping onto Markov networks, they may in reality be using probabilities rather than activations, with this implicit use of probabilities and their well-defined semantics being what enables coherent symmetric processing. Together, this dichotomy and its sub-dichotomy yield a distinction between symmetric, probabilistic processing versus asymmetric, activational processing.

4. Combinatory versus Noncombinatory

The final dichotomy concerns the flexibility with which dyads can be combined, getting at the combinatory aspect deferred from Section 2. In its simplest form, this bears on whether dyads can be combined in arbitrary ways, as presupposed in traditional symbol processing, versus only in a fixed set of ways, as in neural networks.

There are both representational and reasoning aspects to this dichotomy, each of which yields a relevant sub-dichotomy. The easiest way to understand the representational sub-dichotomy is to consider predicate versus propositional logic. The atomic formulas to which truth is assigned in the latter are limited to unitary elements – i.e., symbols such as `A` – whereas in the former they are predicates – or relations – over tuples of terms, such as `Above(id:A value:B)`. Shared elements among relations can also yield structured networks of them. The reasoning sub-dichotomy concerns whether there is a first-order, lifted, or variablized aspect to the processing versus it all being grounded in constants. Canonical symbol systems thus comprise (symbolic) relations with variablized processing. Canonical neural systems comprise (subsymbolic) propositions with grounded processing that is limited by how the elements are interconnected. Combinatory representation and processing may lead to combinatorics, in the traditional sense of computational complexity, but it need not do so, and combinatorics itself is not the focus here.

The proposed central integrating concept here is *lifted dyadic tensors* (Table 2). In such a representation, 0D tensors correspond to propositions, whereas *n*D tensors correspond to *n*-ary tuples. The values along each dimension of a tensor are the symbols – as defined weakly in Section 2 but used more strongly here in Table 2 – that can be in that dimension's position in the tuple, with the locations within the tensor, such as *Red\Square* in Table 2 – or even `(color:Red`

Table 3. 3D map filled in with a range of standard AI approaches.

Combinatory	*Symbolic*	*Subsymbolic*
Symmetric	First-Order Logic, Statistical Relational	Relational Boltzmann Networks
Asymmetric	Rules, Analogical Mapping	Tensors Relational and Graph NNs

Noncombinatory	*Symbolic*	*Subsymbolic*
Symmetric	Probabilistic Graphical, Constraint Propagation	Hopfield and Boltzmann Networks
Asymmetric	Arithmetic Circuits, Sum-Product Networks	Feedforward and Recurrent NNs

`shape:Square)` – serving as instances of the relation. Metadata is associated with each such instance – such as 0.3 for *Red\Square* – yielding a tensor of dyads.

Such a representation can span symbolic relations, neural tensors (Smolensky & Legendre, 2006), distributed vectors (Jones & Mewhort, 2007), semantic pointers (Eliasmith, 2013), and vector symbolic architectures (Levy & Gayler, 2008). It can also represent individual neural units via 0D tensors, or vectors (1D tensors) of units corresponding to whole layers, or even arrays (2D tensors) of units for special cases like vision. When lifted – as in `(color:x shape:Square)` – whole dimensions may be represented by single variables, and sets of dyads that are processed identically may be processed as if they were single elements.

As with the other dichotomies, not all "symbol" systems are combinatory nor are all "neural" systems noncombinatory. For the former, consider arithmetic circuits and sum-product networks, which are both symbolic, but which compute results in time that is linear in the size of their predefined networks. For the latter, consider either the tensor approaches to neural modeling already discussed or convolutional neural networks (LeCun et al., 1989). A convolutional network for vision, for example, defines a tile – i.e., a rectangular subregion of pixels – as an array (a 2D tensor), with a variable origin that lets it align with arbitrary sub-regions of the image.

5. Map of Standard AI Approaches

Table 3 shows the 3D map from Table 1, as filled in with a range of mostly standard AI approaches. Classical symbol systems – including statistical relational ones – do show up at the opposite corner from classical (including deep) neural networks. However, all of the other cells are also filled, albeit some with approaches that have not yet been mentioned and which may be less familiar to cognitive-systems audiences, such as the subsymbolic, combinatory approaches of relational Boltzmann machines (Kaur et al., 2017) and neural networks (Blockeel & Bruynooghe, 2003), and graph neural networks (Gori, Monfardini, & Scarselli, 2005). The work on relational Boltzmann machines was in fact only tracked down after an initial map was created with a blank cell for subsymbolic, asymmetric, combinatory approaches. Other less obvious opposites are also apparent in this more fleshed out map, such as relational Boltzmann machines versus sum-product networks, and constraint propagation versus tensors. What to make of such additional opposites, beyond their nonobviousness, remains to be determined.

This map is almost certainly incomplete as it stands, intended as it is to only be illustrative at this point of the diversity of approaches and their relationships, rather than exhaustive. Left to future work is exploring AI approaches more systematically, including those that may not fit so nicely. Still, what is already there arises from across many of the central paradigms in the history of AI. Also, it is already clear that some of what is missing appears to be non-elemental, in the sense of requiring a combination of approaches from across cells. For example, the map includes analogical mapping in the symbolic, asymmetric, combinatory cell, where information from a source analogue is transferred to the new situation, but the full analogy process also requires a prior retrieval of candidate analogues and selection of one among them, with this other problem more akin to declarative memory access and thus to the symmetric approaches. Similarly, solving constraint satisfaction problems typically requires not only constraint propagation – a symbolic, symmetric, noncombinatory process – but also *conditioning*, a combinatoric search process in which candidate assignments of values to variables are explored when the propagation process fails to resolve all of the ambiguity existing within the problem.

The map also highlights key similarities and differences among these approaches. Consider, for example, how it displays that for every symbolic approach there is a corresponding subsymbolic approach and vice versa. Or consider the L-shaped path from probabilistic graphical models through arithmetic circuits to feedforward networks, which captures the essence of Choi and Darwiche's analysis of the similarities and differences between the endpoints of this chain. The symmetric versus asymmetric dichotomy is also intriguing, in distinguishing pairs of approaches that seem similar but that embody crucial differences that have not always been easy to articulate.

6. Applying the Map to Sigma

Cognitive and AGI (artificial general intelligence) architectures (Kotseruba & Tsotsos, 2018; Goertzel, 2014) may span multiple of the cells in Table 3, although not necessarily replicating all of the approaches listed for them. Consider the Common Model of Cognition, which is the original source of the (sub)symbolic central integrating idea. It also supports forms of both sides of the (a)symmetry dichotomy, in terms of (symmetric) declarative memory and (asymmetric) procedural memory (Rosenbloom, 2019). It further supports combinatory structures, although in an ideally noncombinatoric cognitive cycle – limited to ~50 msec per cycle in people – while enabling combinatoric search across cycles. In this section we apply this map to analyzing the Sigma cognitive architecture – one of the three that most heavily influenced the initial version of the Common Model – along with relevant supra-architectural capabilities that, in conjunction with the architecture itself, begin to form a more complete cognitive system.

6.1 Overall Structure of Sigma

Sigma is structured into two architectures, one graphical and the other cognitive, both of which yield languages within which knowledge and skills can be expressed. The lower-level, graphical

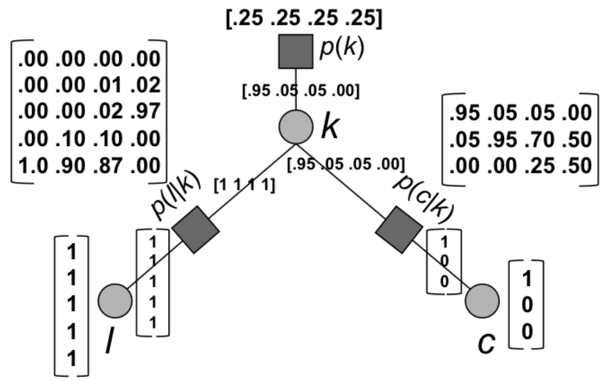

Figure 1. Message passing via the sum-product algorithm for the marginal on the concept (k) in the factor graph for $p(k,l,c) = p(k)p(l|k)p(c|k)$. [Reproduction of Figure 44 from Rosenbloom et al. (2016a).]

architecture is an extended version of factor graphs and the sum-product algorithm (Kschischang, Frey, & Loeliger, 2001; Rosenbloom, Demski, & Ustun, 2016b). Factor graphs are a general form of graphical model that subsumes both Bayesian and Markov networks, and that can be used in solving arbitrary multi-variate functions, not just probabilistic ones. The sum-product algorithm is a general message-passing method for solving graphical models.

The cognitive architecture enables expressing knowledge and skills at a higher level based on a deep combination of ideas from rules and probabilistic reasoning. The graphical architecture, in a sense, implements the cognitive architecture, in that knowledge and skills in the cognitive language are compiled down to graphs that are solved via the extended sum-product algorithm.

In this way, this combination is analogous to a variety of layer pairs in standard computer systems, such as the combination of high-level and assembler languages, or computer and microcode architectures. It is also like Alchemy (Domingos & Lowd, 2009), which compiles probabilistic first-order logic into Markov networks, and Choi and Darwiche's approach of compiling Bayesian networks into arithmetic circuits. Alchemy, however, only supports symmetric symbolic processing, compiling from the combinatory cell of this type to the noncombinatory one, by – unless there is lifting – generating an exponential amount of structure that is processed in time that is linear in the size of the resulting structure. Choi and Darwiche instead compile from the symbolic, noncombinatory, symmetric cell to the corresponding asymmetric one, a process that ends up being complicated by this (a)symmetry mismatch.

In contrast to such approaches, the full table is potentially available in both of Sigma's languages, enabling broad scope with simple within-cell compilation. This, furthermore, occurs in a manner that directly leverages the similarities and differences among the cells, rather than requiring distinct modules for each. The remainder of this section analyzes Sigma in terms of (1) the third hypothesis, by looking at how its two architectures/languages support the three dichotomies, and thus potentially enable at least something in each cell and (2) the fourth hypothesis, by exploring the specific approaches in Table 3.

6.2 Graphical Architecture/Language

As mentioned in the previous section, Sigma's graphical architecture originated with factor graphs and the sum-product algorithm (Figure 1). Factor graphs are undirected bipartite graphs with factor nodes representing the factors into which the overall function is decomposed and variable nodes connecting them based on shared variables. The sum-product algorithm computes marginals over the graph's variables by passing messages between adjacent factor and variable nodes. Because this enables factor graphs to represent arbitrary numeric functions, rather than just probabilistic ones, this initial version could already exploit data/metadata dyads to generically span the (sub)symbolic dichotomy. However, factor graphs are symmetric and noncombinatory, and so by themselves neither support the asymmetric rows in the map nor any of the symmetric combinatory approaches in its top plane.

Asymmetry has been introduced by leveraging the proposed central integrating concept for (a)symmetry; in other words, by allowing selective omission of messages along one direction of what would normally be symmetric links. This does break the clean semantics of factor graphs for asymmetric regions, but still supports them in symmetric regions while providing a key aspect of rules, neural networks, and other asymmetric structures. This approach to spanning (a)symmetry differs from approaches based on distinct modules – whether for Bayesian networks and arithmetic circuits (Choi & Darwiche, 2018) or for declarative and procedural memories (Laird, Lebiere, & Rosenbloom, 2017) – by enabling mixing across the dichotomy at the very fine granularity of individual links in the graph.

With respect to the data-versus-metadata nonlinearity sub-dichotomy, Sigma incorporates two additional extensions to its factor graphs, one for each form of nonlinearity. For data nonlinearity, at the end of each cognitive cycle it can alter the functions in factor nodes that form the Common Model's notion of a working memory for use in the next cycle, corresponding to what standard rule systems do between cycles. For metadata nonlinearity, an approach akin to that in Choi and Darwiche has been leveraged, whereby special-purpose factor nodes may hold nonlinear metadata transforms, whether for rule negation – the original reason for adding this capability – or for the nonlinearity required in feedforward/deep neural networks.

To enable a full combinatory capability, one further extension is necessary: extending the factor graphs to statistical relational by leveraging the (non)combinatory integrating idea of lifted dyadic tensors. Such tensors can express probability distributions over sets of random variables – with one dimension per variable – or delineate the relational instances that match a rule condition. Via tensor products, they can also compute the compatible bindings across sets of rule conditions. The representational aspect of this lifting is supported by enabling consideration of all elements along a tensor dimension, as if a variable existed for that dimension. The processing aspect is supported by enabling each location along a tensor dimension to represent an arbitrary span of symbols along that dimension, as long as the metadata values for that span are within epsilon of each other for each combination of symbols along the other dimensions, thus enabling entire sets of comparable elements to be processed in one step.

Table 4. Example conditionals for a rule, a leg of a naïve Bayes classifier, and a layer of a neural network. (a) Transitive rule for Above predicate with variables connecting conditions and actions. (b) Leg of a naïve Bayes classifier for the conditional probability distribution *P*(#Legs | Concept). (c) Forward input layer of a two-layer neural network. `s` inserts a sigmoid node in the action graph. `v` uses vector rather than probabilistic gradient descent learning.

(a)
```
CONDITIONAL Transitive-Above
    Conditions:  Above(id:a value:b)
                 Above(id:b value:c)
    Actions: Above(id:a value:c)
```

(b)
```
CONDITIONAL Concept-Legs
    Condacts: Concept(object:o value:c)
              Legs(object:o value:l)
    Function(c, l): .9:(table, 4) …
```

(c)
```
CONDITIONAL Layer1-Forward
    Conditions: Input(arg:i)
    Actions: Hidden(arg:h) [s]
    Function(i, h): -1.739:(0, 0
```

6.3 Cognitive Architecture/Language

Sigma's cognitive architecture/language is based on *predicates* and *conditionals*. Predicates have names plus zero or more typed arguments. They define the varieties of dyadic tensors usable by the system, whether (subsymbolic) tensors or (symbolic) relations. Conditionals combine the conditionality found in both asymmetric structures, such as rules and neural networks, and symmetric structures such as graphical models and constraint networks. They consist of combinations of predicate patterns – conditions, actions and *condacts* (which symmetrically blend both conditions and actions) – plus functions. The predicate patterns and associated data types in essence specify the symbols/data and relations, with variables usable for representational lifting, while the functions specify the quantitative metadata.

With just conditions and actions – which asymmetrically match to working memory and change it – rules result, such as the simple one in Table 4(a) (all conditionals shown here are simplified for readability). With just condacts plus functions, fragments of factor graphs result (as in Table 4(b)). Nothing about either of these conditionals demands meaningful data rather than metadata, although data certainly is more critical for the former and metadata for the latter. Likewise, nothing about conditions, actions, condacts, or functions restricts their combinations in conditionals to be purely symmetric or asymmetric. Combinatory processing itself arises directly from variable usage.

Table 4(c) shows a rule-like manner in which the forward portion of a neural-network layer can be encoded via a condition and an action, but with also a function (which, in contrast to the one in Table 4(b), is asymmetric and activational rather than symmetric and probabilistic). In this lifted format, a single conditional with a 1D function can represent a full neural-network layer; although it could instead have been decomposed into a set of propositional conditionals with 0D functions, one for each connection.

6.4 Coverage of AI Approaches

In total, the previous two subsections show how Sigma supports at least one approach from six out of the eight cells in the map, missing at present only the two subsymbolic symmetric ones. Here we explore the second additional hypothesis by assessing Sigma with respect to the specific approaches listed in the table. Most of Sigma's extant capabilities are covered in Rosenbloom, Demski and Ustun (2016a). Only the more recent ones not included there will be accompanied by additional citations in what follows.

The eight cells in the table will be covered systematically by exploring them in a breadth-first fashion – starting from the combinatory, symbolic, symmetric cell at the top left – in order to highlight the incremental changes that occur with movement to adjacent cells. To make this easier to follow, the outer loop of the search will remain on the combinatory plane, with each noncombinatory cell covered immediately after its combinatory partner.

At the origin sit both first-order logic and statistical relational systems. Covering the latter is a direct consequence of Sigma's grounding in factor graphs with lifted dyadic tensors. Table 4(b) showed a simple example of this at the cognitive level. Although first-order logic has not yet been demonstrated in Sigma, following the Alchemy approach of reducing (probabilistic) first-order logic to (probabilistic) graphical models (Domingos & Lowd, 2009) seems a promising approach here. Moving down to the corresponding noncombinatory cell, coverage of probabilistic graphical models is directly implied as a special case of statistical relational systems. Constraint propagation has been demonstrated via variants of the conditional in Table 4(b), with a Boolean function identifying whether or not combinations of variable values are valid. Although this is a lifted form, it is straightforward to ground it instead in a set of propositional conditionals.

To the right are relational Boltzmann machines, the subsymbolic equivalents of statistical relational systems that deal with relational rather than only vector representations; and, moving down to the noncombinatory plane yields the more familiar approaches of Hopfield networks and Boltzmann machines. None of these approaches have been demonstrated to date in Sigma but given the close ties between Boltzmann machines and Markov networks, this may be straightforward.

Starting again from the origin and moving down to its asymmetric neighbor yields rules – as exemplified in Table 4(a) – and analogical mapping. Both have much in common with first-order logic – in particular their symbolic, combinatory natures – but there have long been arguments concerning the appropriateness of (production) rules versus (formal) logic. In Sigma, rules and working memory from the cognitive level map onto separate fragments of the total graph at the level below, with an approach much like the Rete algorithm (Forgy, 1982) implemented via asymmetric graph structures for both condition match and action instantiation. Early experiments on a symbolic, combinatory form of analogy, based on the Structure-Mapping Engine (Falkenhainer, Forbus & Gentner, 1986), have been done in Sigma (Parker, *Personal Communication*, 2013). Although not extremely relevant here, a subsymbolic form of analogy (Mikolov et al., 2013) based on tensors has also been demonstrated.

Moving down to the corresponding noncombinatory cell, Sigma has been shown to be capable of embodying sum-product networks (Joshi, Rosenbloom, & Ustun, 2018), which are effectively asymmetric variants of probabilistic graphical models. Although, when restricted to be complete and consistent, they can only solve a subset of the problems solvable via probabilistic graphical models, they are of particular interest because of their ability to yield polynomial solutions for important problems like probabilistic parsing that are inherently exponential in probabilistic graphical models. The conditionals for sum-product networks turn out to be very much a hybrid among the conditionals used for rules (Table 4(a)), probabilistic graphical models (Table 4(b)) and feedforward neural networks (Table 4(c)) in the three neighboring cells. In particular, they fuse aspects specific to each: rules – symbolic with conditions and actions; probabilistic graphical models – a probabilistic function; feedforward neural networks – use of a function with conditions and actions. As such, they provide one of the more compelling examples of the close relationships that can exist among neighboring cells in the table and thus of the potential benefits of a non-modular strategy for supporting such a space of approaches. We have not yet attempted arithmetic circuits in Sigma but with their close relationship to sum-product networks this should not be difficult.

The last combinatory cell includes tensors plus relational and graph neural networks, combinatory opposites of first-order logic and statistical relational systems but sharing important features with both of their combinatory neighbors (i.e., the other two combinatory cells). Basic tensor processing in Sigma is a direct consequence of the subsymbolic use of lifted dyadic tensors and the operations performed on them by the sum-product algorithm. Relational and graph neural networks have not yet been explored in Sigma and remain tasks for future work. Moving down now to the corresponding noncombinatory variants, feedforward neural networks have been demonstrated (Figure 4) in Rosenbloom, Demski and Ustun (2016b). This has also recently been extended to recurrent neural networks.

Thus, Sigma provides an exemplar of how to integrate across approximately one half of the approaches in Table 3, with plausible stories readily available for the rest. It furthermore does so not by including separate modules for each of these approaches, but instead by attempting to capture their underlying commonalities and differences.

7. Summary and Conclusion

Motivated initially by the goal of understanding better the traditional, coarse-grain dichotomy of symbolic versus subsymbolic that is typically used to distinguish symbolic approaches from neurally-inspired ones, this dichotomy has been deconstructed into three more basic ones: (sub)symbolic, (a)symmetric, and (non)combinatory. A central integrating concept has been proposed for each – dyads of symbolic data and quantitative metadata, direction of dyad flow, and lifted dyadic tensors – as a means of understanding them more deeply and of suggesting ways to ultimately integrate across them. As proved useful during the analyses, additional sub-dichotomies have also been called out for each of these dichotomies.

The cross product of the dichotomies yields a 3D (2×2×2) map that is at the heart of four new hypotheses, two core ones that provide the major results of this article and two additional ones that are more speculative. Each of these hypotheses has been articulated, explored, and evaluated in this article, but to varying extents. The two core hypotheses are:

1. Symbolic/statistical and neural/ML approaches appear at opposite corners of the 3D space that is defined by the cross product of the dichotomies, with the former characterized by the bare terms in each dichotomy and the latter by the prefixed terms.

2. The 3D space frames a novel form of map over a range of mostly standard AI approaches that can help understand both the overall topology of AI and the core commonalities and differences among the approaches that populate the map.

The two additional hypotheses are:

i. Every complete cognitive system must support some approach in each of the map's eight cells.
ii. Every complete cognitive system must support not just some approach in each of the eight cells, but an approach in each cell that captures the essence of what makes it uniquely powerful and has thus justified work focusing on it in isolation.

The two core hypotheses have been explored and evaluated in terms of the structure of AI in general, yielding the beginnings of a 3D map of AI approaches that spans from symbolic/statistical to neural/ML while structuring the relationships among a wider range of AI approaches from different paradigms that, at least according to the map, can be considered as intermediaries between them. The two additional hypotheses have not yet received any form of general evaluation, but a step in this direction has been taken by leveraging them in analyzing the Sigma cognitive system – including the architecture and a number of supra-architectural capabilities – which touches in some manner on many of the cells.

Future work should include investigating additional dichotomies and approaches in order to yield a more complete and informative map of AI approaches, with learning in particular a central component. It should also include understanding how any additional approaches either fit within individual cells, imply aggregates across multiple cells, or do not fit at all. Also important is exploring the extent to which the additional hypotheses, or variations on them, ultimately prove valid.

A critical question that is not relevant to the first two hypotheses, but that is central to the last two, is whether both sides of each dichotomy are truly needed for full intelligence. For example, is there a distinct essence in each cell that must be present separately, or do some parts completely subsume others – such as combinatory subsuming (non)combinatory – in a manner that makes those that are subsumed unnecessary in any explicit form? Even without subsumption, is it possible to construct one approach from another – such as in constructing symmetric autoencoders from asymmetric feedforward networks – so that only one is actually needed in the architecture? Whatever ends up being true, these are interesting and relevant questions to ask, with the discussion here having provided a particular start.

Acknowledgements

This effort has been sponsored by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Ackley, D. H., Geoffrey E. Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*, 147–169.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge, UK: Cambridge University Press.

Blockeel, H. & Bruynooghe, M. (2003). Aggregation versus selection bias, and relational neural networks. *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data* (pp. 22–23). Acapulco, Mexico: University of Massachusetts Amherst.

Cho, B., Rosenbloom, P. S., & Dolan, C. P. (1991). Neuro-Soar: A neural-network architecture for goal-oriented behavior. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 673–677). Chicago, IL: Lawrence Erlbaum Associates.

Choi, A. & Darwiche, A. (2018). On the relative expressiveness of Bayesian and neural networks. *Proceedings of Machine Learning Research*, *72*, 157–168.

Darwiche, A. (2002). A logical approach to factoring belief networks. *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning* (pp. 409–420). Toulouse, France: Morgan Kaufmann.

Domingos, P. & Lowd, D. (2009). *Markov logic: An interface layer for artificial intelligence*. San Raphael, CA: Morgan & Claypool.

Eliasmith, C. (2013). *How to build a brain: An architecture for neurobiological cognition*. New York, NY: Oxford University Press.

Falkenhainer, B., Forbus, K.D., & Gentner, D. (1986). The Structure-Mapping Engine. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 272–277). Philadelphia, PA: AAAI Press.

Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, *19*, 17–37.

Getoor, L. & Taskar, B. (2007). *Introduction to statistical relational learning*. Cambridge, MA: MIT Press.

Goertzel, B. (2014). Artificial general intelligence: Concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, *5*, 1–46.

Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. *Proceedings of the 2005 International Joint Conference on Neural Networks* (pp. 729–734). Montréal, Canada: IEEE.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. arXiv:1410.5401.

Harris, D. & Harris, S. (2012). *Digital design and computer architecture* (2nd ed.). San Francisco, CA: Morgan Kaufmann.

Hart, P. (1982). Directions for AI in the eighties. *ACM SIGART Bulletin*, *79*, 11–16.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedomgs of the National Academy of Sciences of the USA*, *79*, 2554–2558.

Jilk, D. J., Lebiere, C., O'Reilly, R. C., & Anderson, J. R. (2008). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, *20*, 197–218.

Joshi, H., Rosenbloom, P. S., & Ustun, V. (2018). Exact, tractable inference in the Sigma cognitive architecture via sum-product networks. *Advances in Cognitive Systems*, *7*, 39–55.

Jones, M. N. & Mewhort, D. J. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*, 1–37.

Jordan, M. I. & Sejnowski, T. J. (2001). *Graphical models: Foundations of neural computation*. Cambridge, MA: MIT Press.

Kaur, N., Kunapuli, G., Khot, T., Kersting, K., Cohen, W., & Natarajan, S. (2017). Relational restricted Boltzmann machines: A probabilistic logic learning approach. *Proceedings of the Twenty-Seventh International Conference on Inductive Logic Programming* (pp. 94–111). Orléans, France: Springer.

Koller, D. & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. Cambridge, MA: MIT Press.

Kotseruba, I. & Tsotsos, J. K. (2018). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-018-9646-y

Kschischang, F. R., Frey, B. J., & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*, 498–519.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, *38*, 13–26.

Langley, P. Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141-160.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, P., Hubbard, W., & Jackel, L. (1989). Backpropagation applied to handwritten Zip Code recognition. *Neural Computation*, *1*, 541–551.

Levy, S. D. & Gayler, R. W. (2008). Vector symbolic architectures: A new building material for artificial general intelligence. *Proceedings of the First Conference on Artificial General Intelligence* (pp. 414–418). Memphis, TN: IOS Press.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, *8*, 69–97.

McClelland, J. L. & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*, 375–407.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of Advances in Neural Information Processing Systems* (pp. 3111–3119). Lake Tahoe, NV: Curran Associates, Inc.

Minsky, M., Singh, P., & Sloman, A. (2004). The St. Thomas common sense symposium: Designing architectures for human-level intelligence. *AI Magazine*, *25*, 113-124.

Newell, A. & Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, *19*, 113–126.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco, CA: Morgan Kaufman.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532–1543). Doha, Qatar: ACL.

Poon, H. & Domingos, P. (2011). Sum-product networks: A new deep architecture. *Proceedings of the Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 337–346). Barcelona, Spain: AUAI Press.

Rizzuto, D. S. & Kahana, M. J. (2001). An autoassociative neural network model of paired-associate learning. *Neural Computation*, *13*, 2075–2092.

Rosenbloom, P. S. (2012). *On computing: The fourth great scientific domain*. Cambridge, MA: MIT Press.

Rosenbloom, P. S. (2015). Supraarchitectural capability integration: From Soar to Sigma. *Proceedings of the Thirteenth International Conference on Cognitive Modeling* (pp. 67–68). Groningen, The Netherlands: Technische Universität Berlin.

Rosenbloom, P. S. (2019). (A)symmetry × (non)monotonicity: Towards a deeper understanding of key cognitive di/trichotomies and the Common Model of Cognition. *Proceedings of the Seventeenth Annual Meeting of the International Conference on Cognitive Modeling*. Montréal, Canada.

Rosenbloom, P. S., Demski, A., Han, T., & Ustun, V. (2013). Learning via gradient descent in Sigma. *Proceedings of the Twelfth International Conference on Cognitive Modeling* (pp. 35–40). Ottawa, Canada.

Rosenbloom, P. S., Demski, A., & Ustun, V. (2016a). The Sigma cognitive architecture and system: Towards functionally elegant grand unification. *Journal of Artificial General Intelligence*, *7*, 1–103.

Rosenbloom, P. S., Demski, A., & Ustun, V. (2016b). Rethinking Sigma's graphical architecture: An extension to neural networks. *Proceedings of the Ninth Conference on Artificial General Intelligence* (pp. 84–94). New York, NY: Springer.

Rosenbloom, P. S., Demski, A., & Ustun, V. (2017). Toward a neural-symbolic Sigma: Introducing neural network learning. *Proceedings of the Fifteenth Annual Meeting of the International Conference on Cognitive Modeling*. Coventry, UK: University of Warwick.

Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1146–1152). Montréal, Canada: Morgan Kaufmann.

Schuster, M. & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681.

Silver, D., Hubert, T., Anonoglou, J., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters Chess, Shogi, and Go through self-play. *Science, 363*, 1140–1144.

Smolensky, S. & Legendre, G. (2006). *The harmonic mind: From neural computation to optimality-theoretic grammar*. Cambridge, MA: MIT Press.

Sun, R. (2016). *Anatomy of the mind: Exploring psychological mechanisms and processes with the Clarion cognitive architecture*. New York, NY: Oxford University Press

Ustun, V., Rosenbloom, P. S., Sagae, K., & Demski, A. (2014). Distributed vector representations of words in the Sigma cognitive architecture. *Proceedings of the Seventh Annual Conference on Artificial General Intelligence* (pp. 196–207). Quebec City, Canada: Springer.