

SYDE 556/750  
Simulating Neurobiological Systems  
Lecture 10: Symbols and Symbol-like Representations

Andreas Stöckel

Based on lecture notes by  
Chris Eliasmith and Terrence C. Stewart

March 10 & 12, 2020



**Accompanying Readings: Chapter 3, 4 and 9 of “How to Build a Brain”**

# Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Neural Theories of Cognition</b>	<b>1</b>
2.1 Jackendoff's Challenges for Cognitive Neuroscience . . . . .	2
2.2 Solution Attempt 1: Neural Synchrony (Oscillations) . . . . .	2
2.3 Solution Attempt 2: Neural Blackboard Architecture . . . . .	4
2.4 Solution Attempt 3: Vector Operators . . . . .	5
<b>3 Vector Symbolic Architectures</b>	<b>6</b>
3.1 Examples: Binding and Unbinding . . . . .	8
3.2 Circular Convolution . . . . .	8
3.3 Jackendoff's Challenges . . . . .	9
<b>4 The Semantic Pointer Architecture</b>	<b>9</b>

# 1 Introduction



**Note:** So far we have mostly been concerned with representing vectors  $\mathbf{x}$  in populations of neurons. In this lecture we instead think about the representation of *symbols*, leading up to the Semantic Pointer Architecture (SPA).

As discussed at the beginning of the lecture, our ultimate goal is to build models of human brains, and, by extension, human cognition. So far it seems as if we had not made much progress in this direction. What we have discussed so far is “merely” allowing us to represent vectors in a population of neurons, to compute transformations of the represented values within the connections between populations, and to build dynamical systems treating the represented values as control-theoretic state variables.

In particular, one hallmark of human cognition is *language*. So far, it is not clear how to apply the NEF to language, or facts expressed in language, such as “the number eight comes after the number nine”, “all dogs chase cats”, “Anne knows that Bill thinks that Charlie likes Dave”.

We attempt to solve this problem by first having a look at other modelling approaches in Cognitive Science. In particular, we will have a look at so called Vector Symbolic Architectures (VSAs), which can be easily mapped onto the NEF. We then use VSAs to build a particular model of cognition, the Semantic Pointer Architecture (SPA).

# 2 Neural Theories of Cognition



**Note:** For more details on theories of cognition, have a look at Chapter 9 of “How to Build a Brain” [1].

Traditionally, cognitive scientists (and computer scientists) have been working on theories of cognition that involve structured information using some kind of representational framework, such as predicate logic. For example, the first two facts mentioned above could be written in predicate logic like this:

- “The number eight comes after the number nine”: **isSucc**(**EIGHT**, **NINE**).
- “All dogs chase cats”:  $\forall x \forall y (\text{isDog}(x) \wedge \text{isCat}(y)) \rightarrow \text{doesChase}(x, y)$ .
- “Anne knows that Bill thinks that Charlie likes Dave”:

**knows**(**ANNE**, “**thinks**(**BILL**, ‘**likes**(**CHARLIE**, **DAVE**)’”).

Both computer scientists and cognitive scientists have successfully built cognitive models which are roughly based on representations akin to the ones listed above. These “symbolic” approaches are generally quite successful in modelling aspects of human cognition and can match behavioural data. However, they do not answer the question of how these symbols are represented and manipulated within a human brain.

## 2.1 Jackendoff's Challenges for Cognitive Neuroscience

In his 2002 book “Foundations of Language: Brain, Meaning, Grammar, Evolution” [2], linguist Ray Jackendoff poses four challenges cognitive neuroscientists who aim to build a neural model of language. These challenges are

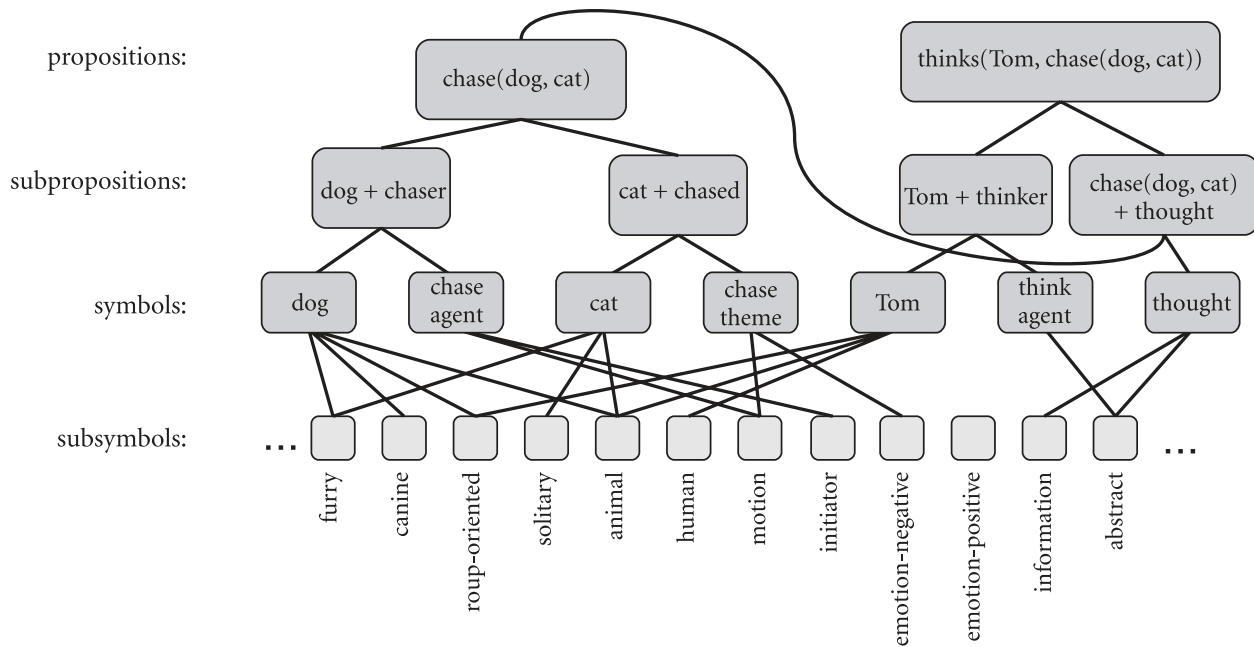
- **The Binding Problem.** Suppose you see a red square and a blue circle. Now, how does the concept of “red” get bound with the concept of “square”, and how is it kept separate from “blue” and “circle”?
- **The Problem of Two.** Consider the sentence “the little star is besides the big star”. How do we keep those two uses of the concept “star” separate? For example, if there was group of neurons representing the concept of “star”, how can this group of neurons both represent the “little star”, the “large star”, as well as the fact that the two stars are next to each other.
- **The Problem of Variables.** Grammar imposes certain rules on sentences. For example, it is “correct” to say “blue x”, if x is a noun, but not “blue y”, if y is a verb. Correspondingly, the question is how these rules, which rely on placeholders, or “variables”, are represented in the brain.
- **Working Memory versus Long-Term Memory.** We can both *use* sentences (working memory; i.e., current neural activities), while also being able to *store* them for very long times (long-term memory; synaptic weights). A neural architecture of language must explain how sentences can be transferred from working memory to long-term memory and back. In other words, we must be able to turn representations from neural activities into synaptic weight changes.

## 2.2 Solution Attempt 1: Neural Synchrony (Oscillations)

One of the earliest theories of how the “binding problem” could be solved is the LISA architecture [3], which has been developed in the early 1990s. This architecture solves the binding problem by proposing to exploit what is known as neural synchrony.

A central idea of this approach is that individual groups of neurons represent concepts—this is a so called *localist* representation; individual spatially colocated groups of neurons correspond to individual concepts. Higher-level concepts can be created by connecting groups of neurons. For example, the neurons representing the symbol “cat” can be connected to neurons representing the concepts “furry”, “solitary”, and “animal” (cf. fig. 1).

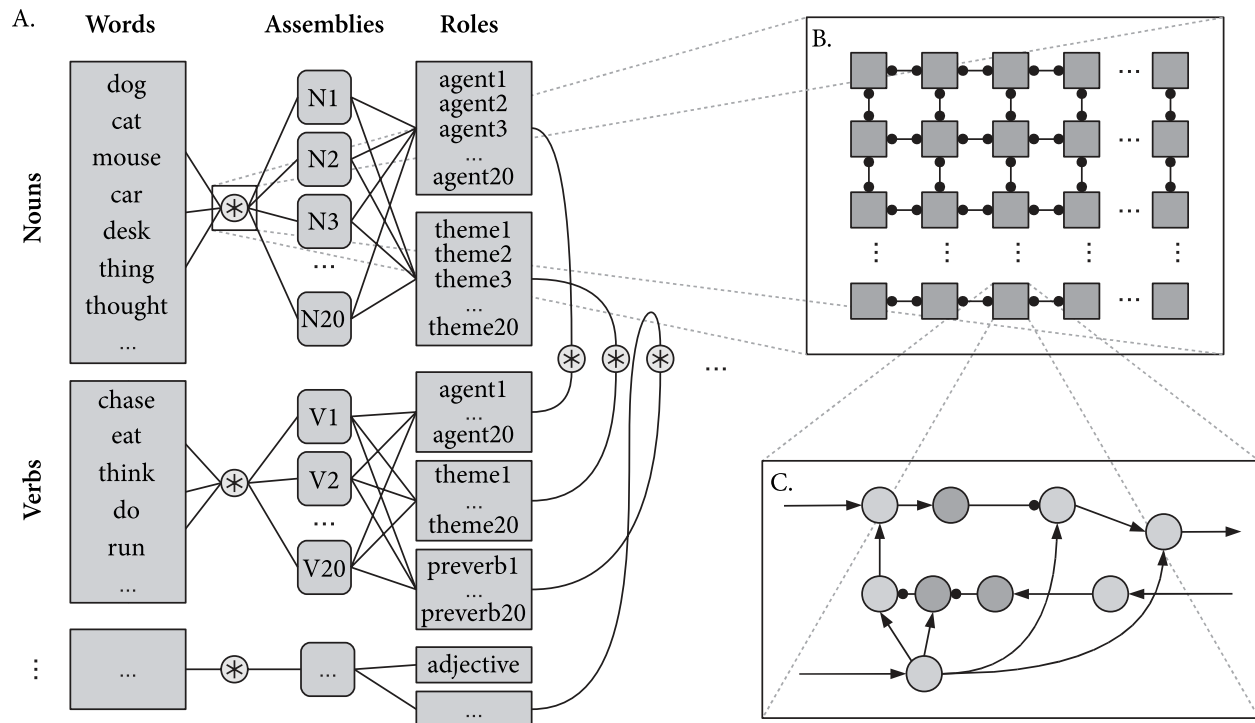
Concepts are bound together by neurons representing the concept oscillating at the same frequency. That is, for our example of “red square and blue circle”, the neuron populations of these concepts would oscillate synchronously in phase, while separate, currently active concepts oscillate with a different frequency and/or out of phase.



**Figure 1:** The LISA architecture. Boxes corresponds to groups of neurons representing individual concepts, lines to synaptic connections. Concepts are “bound” together by neurons oscillating at the same frequency. Figure copied from Figure 9.1 in Eliasmith, 2013 [1], which is in turn adapted from Hummel and Holyoak, 2003 [3].

Let's evaluate this architecture.

- ➕ The architecture solves the binding problem.
- The architecture assumes localist representation, i.e., a few neurons represent a single concept. Most evidence points towards a distributed representation.
- It is unclear how this architecture would solve the “problem of two” – maybe the same symbol used multiple times could oscillate with two superimposed frequencies.
- This architecture cannot solve Jackendoff's third and fourth challenge.
- ➖ It is unclear how these oscillations are generated and controlled in the first place, i.e., how is language translated into activation of these localist representations?
- ➖ Furthermore, it is unclear how the representations are processed—which mechanism is reading out the oscillations and decides which neural ensembles are active together?
- ➖ Finally, we have an exponential explosion of neurons required to represent all kinds of different concepts. For example, assume that there are  $10^4$  symbols. Then, we have on the order of  $10^8$  possible second-order combinations of symbols (“subpropositions”), and on the order of  $10^{16}$  possible propositions. This already exceeds the number of neurons in the brain by a factor of one to ten thousand.



**Figure 2:** Neural Blackboard Architecture. **(a)** Groups of neurons represent concepts (words)—they are connected via a switchboard circuit (symbolised by  $\otimes$  and depicted in panels **(b)** and **(c)**) to slots corresponding to individual roles in a sentence. These roles are then combined via switchboards into higher-level sentences. Image copied from Figure 9.2 in Eliasmith, 2013 [1], which is in turn adapted from van der Velde and de Kamps, 2006 [4].

## 2.3 Solution Attempt 2: Neural Blackboard Architecture

An attempt at solving all four problems posed by Jackendoff is the “Neural Blackboard Architecture” by van der Velde and de Kamps [4], solving the exponential growth problem posed by synchrony approaches. In particular, the idea is to introduce switchboard circuits that can arbitrarily route signals from neurons representing individual concepts to assemblies (a “temporary pool”) representing bound concepts. These high level concepts can then be associated with specific roles a concept might have in a sentence.

- ⊕ This approach has a much lower resource consumption than LISA.
- ⊕ Can solve all four of Jackendoff's challenges (according to the authors).
- ⊕ Explains limitations of human sentence representation.
- This is still a (at least partially) localist representation.
- ⊖ Uses a very particular structure that does not really seem to match biology.
- ⊖ Uses a very large number of neurons; about  $500 \times 10^6$  to represent simple sentences.
- ⊖ Only considers sentence representation, but not the individual control structures.

## 2.4 Solution Attempt 3: Vector Operators

As mentioned, both approaches discussed above are *localist*. We could instead decouple concepts from the underlying neural substrate. That is, populations of neurons can represent different symbols, and send represented symbols for processing to other groups of neurons.

This would be more in line with what the NEF suggests, and knowing that we can represent vectors in neural populations, we could thus represent symbols by assigning vectors  $\mathbf{x} \in \mathbb{R}^d$  to individual symbols. For now, we assume that these vectors  $\mathbf{x}$  have been randomly generated.

This idea is actually quite old, and there have been multiple suggestions as for how to symbol vectors  $\mathbf{x}$  and  $\mathbf{y}$  could be bound together. One approach suggested by Smolensky in 1990 [5] is to simply use a tensor product  $\mathbf{x} \otimes \mathbf{y}$ , a generalisation of a vector outer product to matrices:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{pmatrix} \quad (\text{Outer product})$$

$$\begin{aligned} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} &= \begin{pmatrix} a_{11} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{12} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \\ a_{21} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{22} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \end{pmatrix} \quad (\text{Tensor product}) \\ &= \begin{pmatrix} a_{11} b_{11} & a_{11} b_{12} & a_{12} b_{11} & a_{12} b_{12} \\ a_{11} b_{21} & a_{11} b_{22} & a_{12} b_{21} & a_{12} b_{22} \\ a_{21} b_{11} & a_{21} b_{12} & a_{22} b_{11} & a_{22} b_{12} \\ a_{21} b_{21} & a_{21} b_{22} & a_{22} b_{21} & a_{22} b_{22} \end{pmatrix} \end{aligned}$$

This way, two vectors  $\mathbf{x}$ ,  $\mathbf{y}$  can be bound without losing any information. In contrast to just stacking the two vectors (which would also not incur any information loss), the tensor product has some nice mathematical properties, and can, as outlined by Smolensky in his paper, to a degree be easily implemented in a neural substrate.

- ➕ Solves the binding problem and the problem of two.
- Unclear how to solve Jackendoff's third and fourth challenge.
- ➖ The method scales extremely poorly. Every time two vectors are bound, the dimensionality of the resulting structure is squared; we need  $d^2$  dimensions when binding two  $d$ -dimensional vectors. In general, for  $n$  binding operations we need  $d^n$  dimensions.



**Note:** *Symbolic Architectures and Neuroscience.* All methods discussed so far are trying very hard to map purely symbolic architectures onto a neural substrate. In a sense, neural aspects are treated as *mere implementation details*. This is an instance of the top-down approach we discussed at beginning of the course: mapping high-level cognitive architectures onto biology. In a sense, the hope is that, if successful, neurons would not matter. This is (unfortunately) an assumption many cognitive scientists make.

### 3 Vector Symbolic Architectures

The last idea—using vectors  $\mathbf{x}$  to represent symbols and to then use an operator such as  $\otimes$  to bind them—seems to be reasonable, and fits the Neural Engineering Framework quite well. However, what we would like to have is an operator that maintains the size of the vectors (i.e., takes two vectors of dimensionality  $d$  as an input and outputs a vector of dimensionality  $d$ ), while still allowing us to reconstruct information about the operands.



**Example:** Using “+” as a binding operator. Let’s reconsider the “Binding Problem” as originally posed above. We would like to represent the fact that we are seeing a blue square and a red circle. Hence, we could—for now, randomly—generate four  $d$ -dimensional vector representing these four concepts, where  $d$  is relatively large, for example  $d = 64$ .

To keep us from having to come up with letters  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$ ,  $\mathbf{w}$ , ... for each concept and remember the mapping between letter and concept, we will just write the vector corresponding to each concept in capital letters, like so: BLUE, RED, CIRCLE, SQUARE. Keep in mind that this is just notation; each of these words is a (random),  $d$ -dimensional vector.

Our first attempt at representing the above concept could be to just sum these vectors

$$\mathbf{x} = \text{BLUE} + \text{SQUARE} + \text{RED} + \text{CIRCLE}.$$

However, notice that mere addition of symbol vectors does not allow us to distinguish which colour belongs to which object. Correspondingly, “+” is not a good choice as a binding operator, but may be used to represent that two separate concepts are currently active.

Mathematically, what we would like is an operator  $\otimes$  with the following properties

$$\begin{aligned} \otimes : \mathbb{R}^d \times \mathbb{R}^d &\longrightarrow \mathbb{R}^d, && \text{(preservation of dimensionality)} \\ \mathbf{x} &\approx (\mathbf{x} \otimes \mathbf{y}) \otimes \mathbf{y}^{-1}, && \text{(approximately reversible)} \\ 0 &\approx \langle \mathbf{x} \otimes \mathbf{y}, \mathbf{x} \rangle, 0 \approx \langle \mathbf{x} \otimes \mathbf{y}, \mathbf{y} \rangle. && \text{(dissimilar to inputs)} \end{aligned}$$

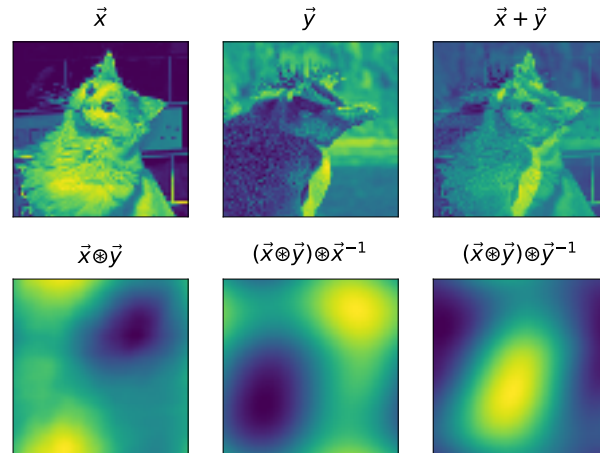
As we have discussed in the above example, the last property ensures that two concepts that are bound to each other cannot be confused with the original concepts. This prevents us from using an operator such as “+” as a binding operator, which preserves similarity, as graphically depicted in fig. 3, and as can be easily shown:

$$\langle \mathbf{x} + \mathbf{y}, \mathbf{x} \rangle = \langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle \approx \|\mathbf{x}\|^2 + 0,$$

assuming that  $\mathbf{x}$  and  $\mathbf{y}$  are two high-dimensional random vectors, which are likely to be orthogonal. We will still use “+” to combine multiple concepts into one symbolic vector.

The use of (random) vectors  $\mathbf{x}$  along with a binding operator  $\otimes$  to construct cognitive symbolic architectures have been called “Vector Symbolic Architectures” (VSAs) by Gayler, 2003 [6]. In the same paper, Gayler argues that such architectures can solve all four challenges put forward by Jackendoff.





**Figure 3:** Similarity preservation and approximate reversibility of circular convolution. The  $+$  operator preserves similarity between  $\mathbf{x}$ ,  $\mathbf{y}$  (both the penguin and the cat are still visible), whereas circular convolution  $\otimes$  does not (neither the penguin nor the cat are visible). Convoluting  $\mathbf{x} \otimes \mathbf{y}$  with the pseudo-inverse  $\mathbf{x}^{-1}$  and  $\mathbf{y}^{-1}$  creates an image that is more similar to  $\mathbf{y}$  and  $\mathbf{x}$ , respectively. The use of images in this example does not imply that we usually apply these methods to uncompressed images. [Code](#)



**Example: Possible Binding Operators.** Among others, the following binding operators have been proposed for use in vector symbolic architectures:

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad (\text{XOR})$$

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \odot \begin{pmatrix} E \\ F \\ G \\ H \end{pmatrix} = \begin{pmatrix} AE \\ BF \\ CG \\ DH \end{pmatrix} \quad (\text{Hadamard Product})$$

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \otimes \begin{pmatrix} E \\ F \\ G \\ H \end{pmatrix} = \begin{pmatrix} AE + BH + CG + DF \\ AF + BE + CH + DG \\ AG + BF + CE + DH \\ AH + BG + CF + DE \end{pmatrix} \quad (\text{Circular Convolution})$$

Circular convolution compresses the outer product by summation along the diagonals:

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \otimes \begin{pmatrix} E \\ F \\ G \\ H \end{pmatrix} = \begin{pmatrix} AE & AF & AG & AH \\ BE & BF & BG & BH \\ CE & CF & CG & CH \\ DE & DF & DG & DH \end{pmatrix} \quad (\text{Outer Product})$$

### 3.1 Examples: Binding and Unbinding

We can use the binding operator “ $\otimes$ ” and the plus operator “+” to encode all kinds of information in a single vector. For example, we can now write the above example as

$$\mathbf{x} = \text{BLUE} \otimes \text{SQUARE} + \text{RED} \otimes \text{CIRCLE}.$$

We can use the reversibility property to “ask questions”. For example, “which object is blue?”

$$\begin{aligned} \mathbf{y} &= (\text{BLUE} \otimes \text{SQUARE} + \text{RED} \otimes \text{CIRCLE}) \otimes \text{BLUE}^{-1} \\ &= (\text{BLUE} \otimes \text{SQUARE}) \otimes \text{BLUE}^{-1} + (\text{RED} \otimes \text{CIRCLE}) \otimes \text{BLUE}^{-1} \\ &\approx \text{SQUARE} + 0. \end{aligned}$$

We can use this technique to write down more interesting concepts, such as the sentences we talked about in the previous section:

- “The number eight comes after the number nine”:

$$\text{NUMBER} \otimes \text{EIGHT} + \text{SUCC} \otimes \text{NINE}.$$

- “The dog chases the cat”:

$$\text{DOG} \otimes \text{SUBJ} + \text{CAT} \otimes \text{OBJ} + \text{CHASE} \otimes \text{VERB}.$$

- “Anne knows that Bill thinks that Charlie likes Dave”:

$$\begin{aligned} &\text{SUBJ} \otimes \text{ANNE} + \text{ACT} \otimes \text{KNOWS} + \text{OBJ} \otimes \\ &\quad \left( \text{SUBJ} \otimes \text{BILL} + \text{ACT} \otimes \text{THINKS} + \text{OBJ} \otimes \right. \\ &\quad \left. \left( \text{SUBJ} \otimes \text{CHARLIE} + \text{ACT} \otimes \text{LIKES} + \text{OBJ} \otimes \text{DAVE} \right) \right). \end{aligned}$$



**Note:** *Graceful degradation.* The information we can pack into a single vector  $\mathbf{x}$  is limited! The larger the number of binding operations, and the number of additions, the lower the precision of operations such as the approximate inverse. While this may sound bad, humans have similar limitations: for example, the nesting depth of sentences, or the number of objects we can keep in working memory is limited. This makes for interesting predictions if we use VSAs to model cognitive phenomena.

### 3.2 Circular Convolution

- “Default convolution operator” to use in conjunction with the NEF
- Exhaustively studied by Tony Plate in the 1990s [7]
- Has the properties demanded above (see fig. 3)

- Definition  $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$ :

$$z_i = \sum_{j=0}^{d-1} x_j y_{i-j \bmod d}.$$

- Alternatively, can use the DFT:

$$\begin{aligned}\mathbf{z} &= \mathcal{DFT}^{-1}(\mathcal{DFT}(\mathbf{x}) \odot \mathcal{DFT}(\mathbf{y})) \\ &= \mathbf{T}^{-1}(\mathbf{T}\mathbf{x} \odot \mathbf{T}\mathbf{y})\end{aligned}$$

- Can be easily implemented in the NEF; three constant linear transformations and a multiplication

### 3.3 Jackendoff's Challenges

- The Binding Problem: See above
- The Problem of Two: Use symbols denoting roles

$$\text{OBJ1} \otimes (\text{TYPE} \otimes \text{STAR} + \text{SIZE} \otimes \text{LITTLE}) + \text{OBJ2} \otimes (\text{TYPE} \otimes \text{STAR} + \text{SIZE} \otimes \text{BIG}) + \text{REL} \otimes \text{BESIDES}$$

- The Problem of Variables: How can we manipulate abstract placeholders?

$$\begin{aligned}S &= \text{RED} \otimes \text{NOUN} \\ \text{VAR} &= \text{BALL} \otimes \text{NOUN}^{-1} \\ S \otimes \text{VAR} &= \text{RED} \otimes \text{BALL}\end{aligned}$$

- Working vs. Long-Term Memory:
  - Working memory: activity of neurons, currently represented concepts  $\mathbf{x}$
  - Long term memory: connection weights, functions
  - Functions can return and modify  $\mathbf{x}$

## 4 The Semantic Pointer Architecture

- Combination of multiple ideas:
  - NEF
  - VSAs (with circular convolution; other binding operators possible)
  - Basal Ganglia/Thalamus/Cortex control loop
  - Theory as for where  $\mathbf{x}$  comes from; *compression* and *decompression*  $\Rightarrow$  Semantic Pointers

## References

- [1] Chris Eliasmith. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford Series on Cognitive Models and Architectures. New York, New York: Oxford University Press, 2013. 456 pp. ISBN: 978-0-19-026212-9.
- [2] Ray Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, USA, 2002. 504 pp. ISBN: 978-0-19-926437-7.
- [3] John E. Hummel and Keith J. Holyoak. "A Symbolic-Connectionist Theory of Relational Inference and Generalization." In: *Psychological Review* 110.2 (2003), pp. 220–264. ISSN: 1939-1471(Electronic),0033-295X(Print). DOI: 10.1037/0033-295X.110.2.220.
- [4] Frank van der Velde and Marc de Kamps. "Neural Blackboard Architectures of Combinatorial Structures in Cognition". In: *Behavioral and Brain Sciences* 29.1 (2006), pp. 37–70. DOI: 10.1017/S0140525X06009022.
- [5] Paul Smolensky. "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems". In: *Artificial Intelligence* 46.1 (1990), pp. 159–216. ISSN: 0004-3702. DOI: [https://doi.org/10.1016/0004-3702\(90\)90007-M](https://doi.org/10.1016/0004-3702(90)90007-M). URL: <http://www.sciencedirect.com/science/article/pii/000437029090007M>.
- [6] Ross Gayler. "Vector Symbolic Architectures Answer Jackendoff's Challenges for Cognitive Neuroscience". In: *Proceedings of the ICCS/ASCS International Conference on Cognitive Science*. ICCS/ASCS International Conference on Cognitive Science. Sydney, Australia: University of New South Wales, 2003, pp. 133–138.
- [7] Tony A. Plate. "Holographic Reduced Representations". In: *IEEE Transactions on Neural Networks* 6.3 (May 1995), pp. 623–641. ISSN: 1941-0093. DOI: 10.1109/72.377968.