On the relationship between syntactic and semantic encoding in metric space language models

Whitney Tabor[1]

[1] *University of Connecticut*
whitney.tabor@uconn.edu

May 28, 2021
To appear in *Journal of Cognitive Science*

Abstract

The relationship between form and meaning is central to the theory of language. Traditionally, syntax and semantics are viewed as two different levels of representation. Based on insights from the intersection of dynamical systems theory and the theory of computation, and guided by linguistic data, I argue that there is only one space, a syntactic-semantic one. I model it here as a stable, countably infinite attractor of an iterated map dynamical system. One advantage of this approach is that it supports a unified treatment of grammatical and ungrammatical processing.

Keywords: *syntax, semantics, dynamical automaton, fractal grammar, dynamical systems theory*

Received 25 June 2020; Revised 17 November 2020; Accepted 12 December 2020
Journal of Cognitive Science 22(2): 135-155, June 2021

1. Introduction

1.1 Standard approach: separate syntax and semantics

It has long seemed reasonable in the theory of linguistics[1] to make a distinction between syntactic representation and semantic representation. Chomsky (1957) noted that (1) is a grammatical English sentence, while (2) is not, even though neither sentence makes sense.

(1) Colorless green ideas sleep furiously.

(2) Furiously sleep ideas green colorless.

Therefore, he argued, a theory that attempts to equate (un)grammaticality with semantic (ill-)/well-formedness is a non-starter. In connection with this view, it is common to note that grammatical and semantic well-formedness are doubly dissociable. The sentences above illustrate well-formed syntax with ill-formed semantics (1) and ill-formed syntax with ill-formed semantics (2) but we can also observe ill-formed syntax with well-formed semantics (3) and, of course, there are many strings like (4) that are both syntactically and semantically well-formed.

(3) Those dog that barking all night lonely.

(4) Some cats that live in high-rises are content.

The term “semantics” has several meanings in the literature. Sometimes it is used to refer to the meanings that are associated with linguistic forms. I will refer to this kind of “semantics" as *semantic content* or *meaning*. For example, the semantic content of “riparian" is *associated with the bank of a river*, the semantic content of “[V]-ed" is *event of type [V] that occurred prior to the moment of speaking*. In formalized treatments of meaning, the semantic content of a linguistic expression can be precisely indicated by a well-formed formula (e.g., Steedman, 2000). I will use the term "semantic form" to refer to

[1]Thanks to Jon Sprouse, the participants in the 2020 SemSpace workshop, and two anonymous reviewers for very helpful feedback.

the structure (in, for example, 2nd-order logic) of such formulas. “Syntactic form", by contrast, refers to the structure assigned by a theory of syntax to a particular linguistic expression when it is interpreted in a particular way. The term, ‘semantics" is also used to refer to a mapping posited by some theories between syntactic form and semantic form. I will say “form → meaning mapping" for this sense.

The standard account thus identifies three important elements: syntactic form, semantic form, and semantic content. Focusing, for the moment, on just syntactic and semantic form, formalized theories of language generally posit a very close parallelism between them, so much that one might wonder if they, in fact, belong to a single encoding stratum. Nevertheless, transformational grammars (e.g., Government and Binding, Minimalism) and most unification-based grammars (e.g., LFG, HPSG, many versions of Construction Grammar, The Parallel Architecture) assign the two to different strata. The main motivations for the separation seem to be (1) language behavior effects a relation between language entities and pragmatic entities or actions so it is natural to assume that the mind has an encoding mirroring each side of this relation; (2) syntacticians have made considerable headway in delineating possible forms while semanticists and philosophers have shed much light on the nature of meanings; the two seem to have different properties; thus it seems natural to posit one representation space for each and a mapping between them; (3) model theory, based on Tarski’s formalization of truth for formal languages (Tarski, 1933), specifies a mapping from forms to meanings; although Tarski’s theory is not generally taken to be a theory of human mental nature, it has formed the basis for models of computer “minds" (e.g., denotational semantics— Gordon, 2012 [1979]) and human minds (e.g., Kamp and Reyle, 1993; Heim, 1983) in which the mental part of the system has both an encoding of form and an encoding of meaning.

Further motivation for positing two separate encodings comes from specific linguistic findings. Jackendoff (2002) argues that non-referential elements like English "do"-support "do" ("Do you want a salmonberry?") and dummy "it" and "there" ("It is difficult to persuade Deirdre."/"There is a moose in the garden") provide evidence for a syntax-phonology map that has

no involvement of a third stratum, “Semantics", where the semantic content of world-linked linguistic elements is encoded. Another kind of argument concerns words, such as *likely* and *probable* (5), that have very similar meanings, but nevertheless show different privileges of combination. Analysts have noted that this pattern can be explained if we posit that *likely* identifies its subject with the subject of its complement clause when its complement clause is infinitival, while *probable* does not take an infinitival complement clause and thus makes no such identification, a distinction in syntactic form.

(5) a. It is likely that Mary will win the race.
b. It is probable that Mary will win the race.
c. Mary is likely to win the race.
d. * Mary is probable to win the race.

Both *likely* and *probable* mean something like *occurring with high probability*, so their meanings alone do not seem to predict their different patterning. Moreover, the rule of subject-equation that applies to (5c) can be expressed in a formula which makes no reference to the specific content of *likely*, and the same rule supports computation of the correct meaning for many other predicates as well, again independent of their contents. Such examples are thus efficiently accounted for under separation of syntactic and semantic strata.

These and related observations have prompted many theorists of language to adopt accounts in which syntax and semantics are represented in two distinct spaces.[2] In the most precisely worked out cases, the syntactic space is populated by symbolic objects generated by combinatorial grammars and there is a mapping from syntactic rules to semantic interpretation rules (e.g., Montague, 1970; Steedman, 2000) that describe combinatorial semantic objects. The topology of both of these spaces is fully discrete, meaning none of its elements form a continuum. In the semantic space, the discrete combinatorial elements are further combined with the meanings of the words to produce

[2]Here, by “space" I mean essentially the same thing as “stratum" and “level" as I have used them above. I’m switching to “space" now because this term is more common in the literature on dynamical systems, which I will discuss shortly.

fully referential semantic objects whose truth can be evaluated with respect to a world. This framework offers a natural account of the judgment that sentences like (1) are syntactically well-formed and semantically ill-formed. In particular, the theory specifies syntactic rules which generate a combinatorial structure for the sentence (supporting a positive syntactic judgment) but semantic constraints make its interpretation compatible with no possible world, or at best with only a rather peculiar world. If we take semantic ill-formedness to be produced by a failure of alignment with the ordinary world, the positing of distinct spaces directly supports the possibility of syntactic well-formedness combined with semantic ill-formedness.

1.2 Alternative approach: a single, metric stratum

Despite the successes of the standard assumptions, I advocate an alternative approach here, in which there is only one space and that space is a complete metric space. For present purposes, an important feature of the completeness is that the space has continuum properties, for these support simultaneously the modeling of gradient phenomena and the modeling of recursive symbolic processes, as I explain below.

My approach, called “fractal grammars", is, in many ways, closely aligned other work that seeks to integrate discrete symbolic characterizations of language structure with systems whose states lie in a continuum—particularly neural-symbolic integration and vector space semantics approaches (e.g., Bowman, 2016; Cho et al., 2017; Coecke et al., 2010; Levy and Goldberg, 2014; Levy et al., 2000; Plate, 1995; Sadrzadeh et al., 2017; Smolensky and Legendre, 2006; Socher, 2014). In a slightly different way, fractal grammars are also closely related to “Dynamic Syntax" (Kempson et al., 2001). In that framework, language interpretations, which are expressed in a higher order logical language composed of well-formed formulas, are formed incrementally as words are processed in order of occurrence. Dynamic Syntax, like fractal grammars, thus also has only one, integrated syntactic-semantic stratum. Recently, Dynamic Syntax has been linked with vector space semantic approaches which are well-suited to model some gradient phenomena (Sadrzadeh et al., 2017). Nevertheless, despite, these similarities, there are

some notable differences between fractal grammars and nearly all of the approaches just mentioned (Cho et al., 2017, is a partial exception): fractal grammars lie in metric spaces, not vector spaces; these metric spaces are the state spaces of feedback dynamical systems, and fractal sets play a central role. In General Discussion, I come back to these differences and clarify why I think this alternative approach is worth considering.

As a foundation for making this case, I review several challenges faced by the classical model:

> Challenge 1: Semantics without syntax. Syntax, as standardly understood, uses inviolable symbolic rules to form the constituent structures on the basis of which semantics computes meaning. Therefore, for (3) above, the language system should fail to compute a meaning. Why, then, do we easily make sense of the sentence, while clearly finding it ungrammatical?
>
> Challenge 2: There seem to be degrees of semantic ill-formedness. For example, (6) is semantically odd, but not quite as odd as (7).
>
> (6) Fred drank the cup of ball bearings.
>
> (7) Fred drank the tree.
>
> In the second case I take Fred to have swallowed the tree by gulping with no chewing, and, not for example, to have used a grinder to dissolve the tree into a liquid form ahead of drinking. Classical semantic models (e.g., Montague, 1970) are truth-conditional, and truth is taken to be binary-valued, so, in those models, it is not clear how to model gradience.
>
> Challenge 3: There seem to be degrees of syntactic ill-formedness. For example, Sprouse et al. (2016); Villata et al. (2020) found, in grammaticality judgment experiments on extraction of constituents from linguistic islands, that all the islands they tested showed tell-tale signs of grammatical compromise, but the effect sizes were weaker in some than others (8).

(8) a. Which puzzle did you think that the candidate solved? [GRAMMATICAL]

b. Which puzzle did you wonder whether the candidate solved? [ISLAND - WEAK]

c. Which puzzle did you smile because the candidate solved? [ISLAND - STRONG]

Again, because the classical models employ symbolic structures, it is not obvious how they can address syntactic gradience.

Challenge 4: Linguists have long noted that syntactic violations usually produce a more severe sense of badness than semantic ones. This broad intuition, however, may be skewed by frequent experience with syntactic violations which produce an uninterpretable string and correspondingly frequent experience with semantic violations which are interpretable. However, several carefully designed experiments in which all stimuli are easily interpretable still find a stark magnitude difference which arguably corresponds to the semantics/syntax divide (Keller, 2000; Sorace and Keller, 2005; Villata, 2017; Villata et al., 2016). For example, Sorace and Keller's "soft constraints" include definiteness, verb meaning, and referentiality (arguably semantic) while their "hard constraints" include violations of subject-auxiliary word order, of subject-verb agreement, and of resumptive pronoun constraints (arguably syntactic). Villata et al. find that interference effects from "criterial features" which govern syntactic movement produce much more disturbance than interference effects from "non-criterial features" which involve semantic differences in noun meanings.

Challenge 5: In sentence processing, difficulty produced by ill-formedness at one point in a sentence (e.g, in garden-path sentences) often manifests over several subsequent words (so-called "spillover effects"). Under the standard model the relevant ill-formedness is usually associated with a single failure of symbolic

unification. Thus, it is not clear why the effects of ill-formedness should appear also on other words than the one that produced the conflict.

1.3 Discrete update dynamical system

To address these challenges, I will employ a dynamical system. Let $\mathbf{x}$ be a point in a complete metric space, X. A nonautonomous discrete update dynamical system is a function $f : X \rightarrow X$ that updates $\mathbf{x}$ and possibly receives input from an environment simultaneously:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), t) \tag{1}$$

Here, the variable t ("time") starts at 0 and gets incremented by 1 at each iteration of the function. $\mathbf{x}(t)$ gives the state of the system at time t, and $\mathbf{x}(t+1)$ gives the state of the system one time step later. It is natural to think about this system as a brain that may get input from the world (the dependence of f on t supports this) and also may also, simultaneously, evolve its mental state by internal cause (the dependence of f on $\mathbf{x}$ supports this). I will use this system to model the word-by-word processing of sentences (as in speaking, listening, and some forms of reading). At each time step, one word arrives. Note that the dependence of $\mathbf{x}$, through f, on its previous value is a case of "feedback dynamics".

Including feedback dynamics supports two other relevant features: (i) the system may organize itself around attractors, i.e., sets to which it returns when it is displaced ("perturbed") up to some positive radius; (ii) such attractors may be fractal sets. Informally, fractal sets are sets that exhibit a particular complex form at arbitrarily small scales. They are useful because they support modeling recursive combinatorial structures, and these are arguably fundamental to language form and meaning (Montague, 1970). Moreover, the employment of attractors offers a new approach to the syntax/semantics distinction. Specifically, there is a *grammatical manifold*, a proper subset of the space X on which the system travels when it is processing perfectly grammatical input. This manifold is an attractor, so if the system gets knocked

off the manifold by a not-too-large amount, it will come back onto it. Encountering a semantically or syntactically ill-formed word knocks the system off the manifold, and it generally takes it several steps to return. In the case of semantic ill-formedness, the errant word only knocks the system a small distance off the manifold, near enough that, even though it may take a few steps to recover, the model never makes a category error. Syntactic ill-formedness knocks the system much farther off the manifold. In this case, it still eventually returns, but for a time, it visits parts of the space that are out of synch with the sequence of words—in this case, it is making category errors (expecting words of certain types, and getting words of completely different types). Because everything is occurring in a metric space, there is a natural way to model gradience: it is a function of the (real-valued) distance of the system state from the grammatical manifold. Moreover, there is only one space, so the challenges associated with the two-space system described above are avoided.

Here, I explore this new way of thinking about form and meaning through some simple formal examples worked out in the framework of *fractal grammars* (Tabor, 2000, 2003, 2009, 2015), an approach to computing with dynamical systems (Moore, 1998; Siegelmann and Sontag, 1994; Siegelmann, 1999). *Fractal grammars* support combinatorial computation in a complete metric space.

Section 2 reviews fractal grammars. Section 3 explores the novel perspective on the syntax-semantics distinction. Section 4 concludes.

2. Encoding of syntactic and semantic structure

2.1 Formal Languages

To be clear about syntactic structure, it is useful to consider formal languages. A standard approach is start with a finite alphabet of symbols, $\Sigma = \{a_1, a_2, \ldots, a_k\}$ for k a positive integer. Though it is standard to refer to Σ as an “alphabet", I will refer to it here as a “vocabulary" since this usage is more typical in discussions of natural languages. A language $\mathcal{L}$ is indicated by a characteristic function, L, on finite strings drawn from the vocabulary—

the function has value 1 if its input is a grammatical string of the language and has value 0 otherwise. Here, for convenience, we include one-sided infinite strings as well as finite strings:

$$L : \Sigma^{\infty} \rightarrow \{0, 1\} \tag{2}$$

Languages of this form come in two main types, computable and non-computable (Siegelmann, 1999). It is useful (e.g., Kozen, 1997) to organize the computable languages via the *Chomsky Hierarchy* which identifies a series of classes of formal languages, each successive class including the members of the previous class as well as additional languages. A key breakpoint occurs between the finite-state languages, which occupy the lowest rung on the Chomsky Hierarchy and the context-free languages, on the next rung up. The context free languages are the simplest class bearing the property that a computer for recognizing one of them must be able to distinguish an infinite number of states. A very simple form of computer, called a *Pushdown Automaton* can recognize any context free language. The key mechanism in a pushdown automaton is a storage device called a *pushdown stack*. The pushdown stack holds a sequence of symbols in a kind of infinitely extendable tube such that, at any point, only the last symbol is available for manipulation. Pushdown stacks are useful for keeping track of the complex phrasal embedding structure that is typical of natural languages, and virtually every precisely formalized theory of syntax uses them or an equivalent.

It is useful, at this point to consider some neural network technology. Neural networks are a class of dynamical systems which have proven useful for modeling a number of psychological functions.

2.2 Recurrent neural networks for symbol processing

Discrete update recurrent neural networks (RNNs) with real- (or complex-) valued units are dynamical systems of the form (1). For present purposes, I will employ them as symbol processors. A standard strategy (Elman, 1991) is to specify a layered network. The input space encodes the words of Σ as unique one-hot vectors—all zeros except for a single one. These map

forward to one or more hidden layers, at least one of which has recurrent connections. The last hidden layer maps to an output layer, which has the same dimensionality as the input layer—i.e., one unit for every member of the vocabulary. The idea is to configure the weights of the network in such a way that it implements the function L. To set the stage for this, we assume that words of sentences of $\mathcal{L}$ are presented to the network one at a time, in sequence. After each word is presented, the activation of the output layer is computed. We require that at every juncture between words in $\mathcal{L}$, all and only the possible next-words from $\mathcal{L}$ are activated on the output layer. In order to perform such a task successfully, the neural network has to implement a sufficiently powerful processor to handle the syntax of language $\mathcal{L}$. For example, if the language were a context free language, but the network was only capable of inhabiting a finite number of states, it would fail at the task.

It is not hard to see that, for a given finite state language, if one builds a network with enough hidden units, a system with these properties can be set up to implement L for any finite state language. Without going into detail, it is fairly easy to make a recurrent network that activates (depending on appropriate input) a unique hidden unit corresponding to each of finitely many states. The hidden-to-output weights can then map from these finite states to next-symbol possibilities on the output layer. But for infinite state languages—e.g., context free languages—a more efficient use of state information is needed. The reason is that, since all information mediating between input and output must pass through the hidden layer, the hidden layer needs to instantiate infinitely many distinct patterns. It certainly will not work to adopt the strategy used for the finite state model just described of having each hidden unit either be on or off—for N hidden units, there are only 2^N activation patterns of this form. The only option is to distinguish different degrees of activation of the units. Therefore, we assume the units take their activations in a real interval of positive length (e.g., (0, 1)), yielding no shortage of states. However, to finitely specify a computational procedure that will actually work using an infinity of gradations of values of hidden activations, we need some way of organizing the states systematically. Fractal sets provide a way (Tabor, 2000).

2.3 Fractal grammars

I take a *fractal* to be a set whose Hausdorff dimension exceeds its Lebesgue covering dimension. This definition picks out sets that have a complex pattern that repeats itself at arbitrarily small scales.[3]

Natural language syntax appears to have a recursive character that, up to our current empirical detection, can be weakly generated by a context free grammar and structurally specified by a mildly context-sensitive grammar (Savitch, 1987). Consider a discrete-update, one hidden-layer recurrent neural network, $RNN1$, that always starts, at the beginning of processing, with its hidden layer in an initial state x_0 in metric space X and jumps from point to point as specified by equation 1 where the inputs that are indexed by the time parameter t are grammatical in a language, $\mathcal{L}$, which we wish to model with the network. Let G be the set of points in X that are visited by the network during the processing of all possible sequences of grammatical sentences. We refer to G as the *grammatical manifold* associated with $\mathcal{L}$ under $RNN1$. To accurately model $\mathcal{L}$ in the prediction sense specified in the previous section, G must be structured so that each state contains accurate information about possible future transitions, given possible future inputs. The core idea of fractal grammars is to encode the recursively structured information in the symbol sequences of $\mathcal{L}$ by specifying a fractal G.

Let X be a complete metric space.[4] For N a positive natural number, let $f_i : X \to X$, $i \in 1, \ldots, N$ be a set of maps on X with domains, $d_i \subseteq X$, respectively. A *dynamical automaton*, DA, is such a set of maps combined with a single point, $x_0 \in X$, called the *initial state*:

$$DA = (X, \{f_i : i \in \{1, \ldots, N\}\}, x_0) \tag{3}$$

A dynamical automaton, DA, defines a formal language $\mathcal{L}_{\mathcal{DA}}$ as follows.

[3]There are many ways of defining fractals, so I adopt this definition tentatively. This definition, due to Mandelbrot, highlights so-called mono-fractals which have the same pattern at all scales. It may turn out, for example, that for some natural language modeling, multi-fractals, with structure that varies in systematic way across scales, are more suitable.

[4]A metric space is a topological space in which a distance metric is defined. A metric space is complete if all Cauchy sequences converge to points in the space.

Consider the one-sided infinite string, $\sigma = \sigma_1\sigma_2\ldots$ where $\sigma_j \in \{1,\ldots,N\}$ for $j \in 1,2,\ldots$. If, for every $j \in 1,2,\ldots$, the point $x_j = f_{\sigma_j}(x_{j-1}) \in d_{j+1}$, then σ is a member of $\mathcal{L}_{\mathcal{DA}}$. For a finite string σ with final symbol σ_K, if the previous condition holds for all symbols in the string prior to σ_K and $f_{\sigma_K}(x_{k-1})$ is not in any of the d_i's, then σ is a member of $\mathcal{L}_{\mathcal{DA}}$. In other words, $\mathcal{L}_{\mathcal{DA}}$ is the language of allowed sequences of function applications of DA, given the domain restrictions of the f_i's.

As in the case of a recurrent neural network as discussed above, the *grammatical manifold of a dynamical automaton*, DA, is the set of points in X that is visited by DA under all function applications specified by strings of $\mathcal{L}_{\mathcal{DA}}$ starting from the initial state x_0. A *fractal grammar* is a dynamical automaton whose grammatical manifold is a fractal.

2.4 An example fractal grammar

We consider the non-finite-state phrasal embedding language shown in Table 1. This language can be modeled by a pushdown automaton with three stack symbols, which I'll call A, B, and C. Whenever an a occurs, the automaton pushes A onto the stack, and correspondingly for b and c. When a d occurs, the automaton removes the last element from the stack, making visible the item below it. In all cases, knowledge of what symbol is on the top of the stack provides exactly the information needed to tell which words can come next in the sequence.

To process all strings in $\mathcal{L}_{\mathcal{ABCD}}$ the set of stack states that the system needs to distinguish are exactly the members of $\{A,B,C\}^*$.[5] In other words, there needs to be a distinct hidden unit state for each member of this (countably infinite) set. In support of this, Figure 1 shows a fractal-based system for assigning hidden states to stack states. Given this assignment scheme, we can define a dynamical automaton, DA_{ABCD} whose language is $\mathcal{L}_{\mathcal{ABCD}}$ by the function specifications in Table 2.

[5] Σ^* for Σ an vocabulary specifies the set of all finite-length strings that can be formed from Σ, including the empty string. Σ^∞ specifies the set of all infinite-length strings of these symbols.

S → A B C D
A → a (S)
B → b (S)
C → c (S)
D → d

Table 1. The grammar of a pushdown automaton language, $\mathcal{L}_{\mathcal{ABCD}}$, that cannot be generated by a finite state machine. The grammar specifies sentences by symbol replacement. A rule of form "$M \rightarrow D_1 D_2 \ldots D_K$" means "Replace M with the sequence $D_1 D_2 \ldots D_K$". Replacement proceeds until no more replacement is possible. The resulting string is deemed a grammatical sentence. Parentheses specify optionality, so, for example, "A → a (S)" means "'A' can be replaced by 'a' OR 'A' can be replaced by 'a S"'. An example sentence of this language is [a b [a b c [a b c d] d] c [a b c d] d] (Square brackets highlight the recursive embedding but are not part of the string.)

At the beginning of each sentence, the processor starts at $\binom{1/2}{1/2}$ (corresponding to empty stack in a pushdown automaton) and returns to this point when a sentence has been parsed. It cycles on trajectories of the form lower-right → lower-left → upper-left at different scales for the processing of sequences at different levels of embedding.

An advantage of fractal grammars is that they can be implemented in certain types of neural networks (e.g., ones with gating units— Tabor, 2000) and their method of implementing recursion appears to be closely related to the methods induced by widely used learning neural networks (Tabor, 2011).

2.5 Instability of affine fractal grammars

One problem with the fractal grammars described so far, however, is that they lack asymptotic stability. An attractor of a dynamical system is *asymptotically stable* if trajectories that start out near the attractor converge on it as time goes to infinity. In several senses, it is desirable to work with systems that have asymptotic stability. First, the behavior of an asymptotically stable system is strongly governed by the properties of the attractor; this can make it easier to

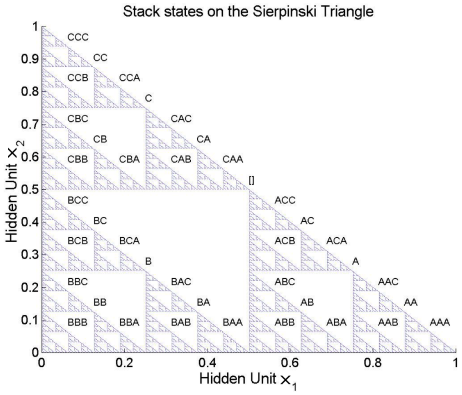

Figure 1. A map from the members of $\{A, B, C\}$* to middles of hypotenuses of the Sierpinski Triangle. The horizontal and vertical dimensions correspond to unit activations in the hidden layer of a neural network for processing $\mathcal{L}_{\mathcal{ABCD}}$.

understand the system. Second, a system with asymptotic stability is reliable: as long as it is not perturbed too far away from the relevant attractor, it will consistently exhibit the dynamics associated with the attractor. Third, human language processing by native speakers shows signs of being asymptotically stable with respect to an attractor associated with grammatical processing. Focusing first on language comprehension, though a language perceiver may encounter various disturbances in the form of noise that interferes with detecting the signal—an unfamiliar word, a syntactically difficult sentence, a garden path sentence, or some other disturbance—such events do not usually flummox the language system for very long. After hearing a few more words, the person is generally back on track with understanding what is being said.[6].

[6]It it is true that a person in this circumstance may persist in failing to understand what is being talked about if, for example, a very important word has been missed. However, this rarely results in failure of the language processing system itself: the person continues being able to parse upcoming sentences. What is disturbed is not the person's language interpretation ability, but their understanding of the relation between the language and the situation at hand.

Function	Domain
$f_b(\mathbf{x}) = \mathbf{x} - \binom{1/2}{0}$	$x_1 > 1/2$ and $x_2 < 1/2$
$f_c(\mathbf{x}) = \mathbf{x} + \binom{0}{1/2}$	$x_1 < 1/2$ and $x_2 < 1/2$
$f_d(\mathbf{x}) = 2\left(\mathbf{x} - \binom{0}{1/2}\right)$	$x_1 < 1/2$ and $x_2 > 1/2$
$f_a(\mathbf{x}) = \frac{1}{2}\mathbf{x} + \binom{1/2}{0}$	X

Table 2. A dynamical automaton DA_{ABCD} whose grammatical manifold is the stack map in Figure 1. The metric space X is the 2-dimensional plane of points (x_1, x_2) with $x_1, x_2 \in \mathcal{R}$ and Euclidean distance metric The initial state is $\vec{x} = \binom{1/2}{1/2}$.

Turning to language production, a native speaker's ability to speak may be temporarily compromised by an interruption, a powerful realization, or even a self-generated verbal confusion, but this situation rarely lasts long: even after experiencing such a difficulty, the speaker will generally continue talking.

What happens if a disturbance is introduced into the fractal processor described above? Figure 2a shows the effect of one small perturbation that occurred on a single word of otherwise perfectly grammatical language. Evidently, a small perturbation produces a very long-lived disturbance—there is no sign in the numerical data of asymptotic reconvergence. A useful way of assessing the stability of an attractor in a deterministic dynamical system is to compute its *Lyapunov exponents*. The Lyapunov exponents measure the average rate of expansion/contraction of the space around the attractor. If a dynamical system has no negative Lyapunov exponents, the system cannot be asymptotically stable. Tabor (2002) defines a natural extension of the definition of Lyapunov exponents to non-deterministic systems with probabilistic transitions. Under this definition, if we consider a version of $\mathcal{L}_{\mathcal{ABCD}}$ in which there is no probability mass on strings with infinitely deep embeddings, then the Lyapunov exponents of the system are all 0. Thus, the system is not asymptotically stable (Tabor, 2002).

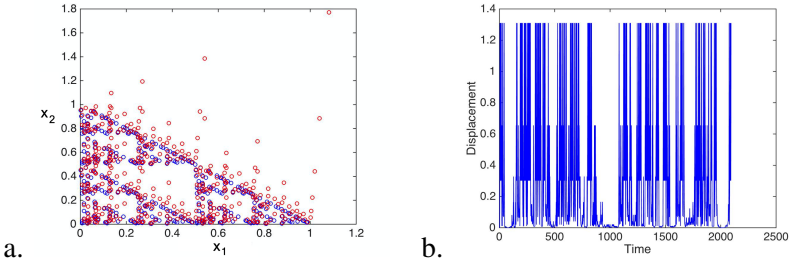

Figure 2. a. Blue circles: A sample grammatical trajectory. Red circles: A trajectory derived from the grammatical trajectory by introducing a small displacement after a few steps of grammatical processing and then continuing to drive the system with the grammatical string thereafter. b. Displacement from the grammatical trajectory versus time.

2.6 Stable dynamical automata

To prepare the ground for building stable dynamical automata, Tabor (2015) introduces the notion of a manifold *labeling*: a labeling is a map from each point on the manifold to a subset of the vocabulary, Σ. The labeling specifies which symbols are allowed under grammatical processing from each point. Suppose a labeled dynamical automaton is perturbed up to some positive radius from its grammatical manifold and then driven forever by a grammatical sequence of symbols. If it always converges back to the grammatical manifold in such a way that, eventually, the next symbol read is consistent with the labeling of the nearest point on the manifold, then the system is said to exhibit *Back in Kansas* stability.[7] In other words, a system with Back in Kansas stability has the ability to recover from perturbations: as long as it gets a long enough string of grammatical input, it will return to properly processing the language.

[7]After returning from her adventures in Oz, Dorothy seemed to settle back into the ordinariness of life in Kansas, once sufficient time had passed.

Tabor (2015) shows that stable dynamical automata exist for some context free languages. If stable dynamical automata can model natural languages, I suggest that they offer a new way of understanding the syntax-semantics relation. The next section explores this idea by examining a specific case.

3. Syntax and semantics in fractal grammar models

As in Tabor (2015), I focus on a mirror recursion language. Mirror recursion refers to languages whose core recursive structure has the form of a grammar like that in Table 3. Corballis (2007), argues that mirror recursion is a key type of recursion in natural languages. For example, English object relative clauses show a mirror recursive pattern with respect to subject verb agreement (9).

(9) a. The cats howl.
b. The cat howls.
c. * The cats howls.
d. * The cat howl.
e. The cats who the girl chases howl.
f. The cat who the girls chase howls.
g. The cats who the girls chase howl. etc.

S → a (S) b
S → x (S) y

Table 3. A grammar that generates the mirror-recursion language, MR1. An example of a string of MR1 is [a [x [a [a b] b] y] b] (square brackets indicate embedding relationships). If we let a = N[Sg], b = V[Sg], x = N[Pl], and y = V[Pl], then the grammar specifies the possible verb agreement patterns in English center-embedded object relative clauses, some of which are illustrated in (9).

To explain the proposed new approach to syntax and semantics, I will work with the language $\mathcal{MR}1$ specified by the grammar of Table 3. In Figure

3, the transitions indicated in bold blue font specify a dynamical automaton, DA_{MR1}, for $\mathcal{MR}1$. By contrast, when all of the tra-nsitions in Figure 3 are considered (bold blue and normal font black), the system becomes a dynamical automaton for the language $\{a, b, x, y\}^{\infty}$. In the present case, we are interested in strings which have a finite series of transitions which are grammatical under DA_{MR1} (bold blue font in Figure 3) followed by a finite series of transitions that may include some steps that are not permitted DA_{MR1} (normal font black), followed by a concluding, infinite series of grammatical transitions. In dynamical systems terms, when the middle segment includes ungrammatical transitions, it produces a perturbation of the trajectory away from the grammatical manifold. Figure 4 shows an example of the system's behavior under a fairly strong perturbation produced by replacing one grammatical transition with an ungrammatical transition in an otherwise grammatical string. In this case, unlike the example described in section 2.5, the effect of the perturbation dies out after 6 time steps. In fact, this system is globally Back in Kansas stable: no matter where it is perturbed to, a sufficiently long sequence of grammatical symbols will eventually bring it back onto the manifold (Tabor, 2015).

Perturbations of this system can be divided into two types. Suppose the system is processing a grammatical string of a's, b's, x's, and y's. At some point, t, in the processing, a rogue word, $w_{semanom}$, is inserted in place of whatever (grammatical) continuation was about to occur and then the sequence continues as before (having skipped the word that would have occurred where the rogue word was). If, as I assume here, $w_{semanom}$ only displaces the processing a small amount—within a radius, $r_{semanom}$, whose magnitude is contingent on where on the grammatical manifold the processor would have landed had $w_{semanom}$ not occurred—then the effect on future processing is minimal: the model will follow a path that has the same future expectations as a corresponding grammatical sentence that had a normal word in place of the rogue word. In this case, we say the model has experienced a *f(ractal)g(rammar)-semantic anomaly*. This is the first type of perturbation. However, if at t, a different rogue word, $w_{synanom}$ is presented and $w_{synanom}$ displaces the state sufficiently far away, then the system will, for one or more

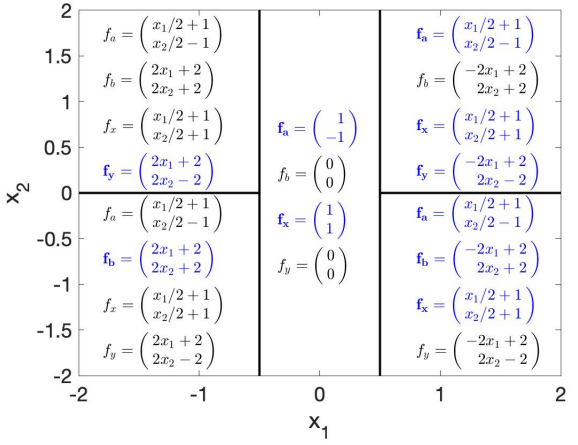

Figure 3. Bold blue maps: DA_{MR1}, a dynamical automaton for processing $\mathcal{MR}1$. The space, X, is the plane with Euclidean distance metric. The initial state is $\binom{0}{0}$. The dark lines indicate a partition of the plane. The two vertical partition boundaries extend to positive infinity and negative infinity. The left horizontal partition boundary extends infinitely to the left, while the right one extends infinitely to the right. The update functions within each partition compartment specify the state update associated with each possible symbol emission for states in the compartment. This system is Back in Kansas stable under any perturbation. All maps: when both bold blue and normal-font black maps are included, perturbations from the grammatical manifold of DA_{MR1} can be introduced by having the system process words in an incorrect order.

words in the future, make erroneous predictions about which continuations are grammatically possible, where by "grammatically possible", I mean possible had the replacement by $w_{synanom}$ not occurred. In this case, we say the system has experienced a *fg-syntactic anomaly* and we say that some *ungrammatical processing* has occurred. In both cases, if the system is Back in Kansas stable, then it will eventually converge back onto the grammatical manifold and it will not make any incorrect predictions after it has done that.

The main question of interest here is whether fg-semantic anomaly and

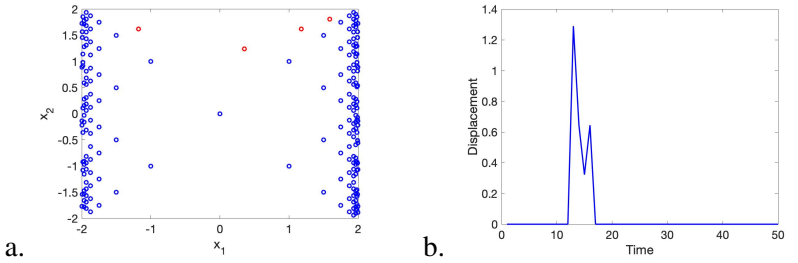

Figure 4. Response of DA_{MR1} to perturbation. a. Grammatical manifold (blue) and perturbed trajectory (red). b. The difference between the perturbed trajectory and the corresponding grammatical trajectory over time.

fg-syntactic anomaly bear a resemblance to natural language semantic and syntactic anomaly respectively. I will consider these questions shortly. Before doing so, it is helpful to address some foundational issues.

First, does each point on the grammatical manifold have an associated $r_{semanom} > 0$? If so, then grammaticality, and the attendant notion, fg-semantic anomaly, have a kind of local robustness–there is a mild level of disturbance that each sentence can tolerate without any future transitions being ungrammatical (i.e., all transitions of the bold blue type in Figure 3). The answer is "yes": this can be shown for DA_{MR1} via the method of inverse iteration described in Tabor (2009). The Appendix sketches a proof. Here, I provide a graphical demonstration of the core insight. First, consider the partition indicated by the dark lines in Figure 3. Each compartment is labeled with its possible next-symbols in bold. Now consider two points, s_{gram} and $s_{semanom}$ corresponding to the grammatical state and the rogue state at time t. If, going forward in time, some grammatical trajectory of the system causes the future of s_{gram} and the future of $s_{semanom}$ to be on opposite sides of the decision boundaries in Figure 3, then the rogue trajectory can give rise to an erroneous prediction at that point. This would violate the definition of fg-semantic anomaly. Therefore, to find out which points around the grammatical manifold will not produce this effect, we can iterate the boundary points under the inverse of the system to find out all the

places where they separate points from one another. Figure 5 shows the result of this test. Indeed, the backward iterations of the boundaries never land on points on the manifold and they form a partition such that each compartment contains at most one point of the grammatical manifold. This indicates that fg-semantic anomaly is a possibility at every word.

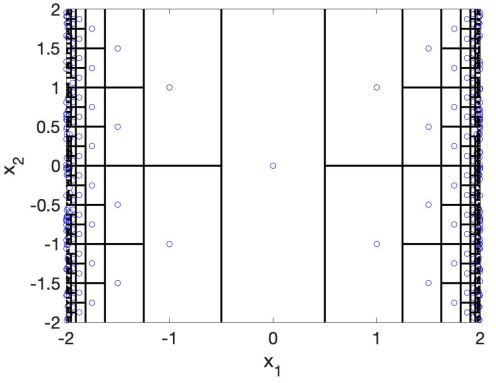

Figure 5. The grammatical manifold (blue circles) along with the image of the decision boundaries (black lines) under the inverse of DA_{MR1} (where it is invertible). Every point on the grammatical manifold lies strictly inside a distinct compartment of the partition defined by the inverse image.

What about fg-syntactic anomaly. Does it exist? The answer is yes because all that $w_{synanom}$ must do to produce fg-syntactic anomaly is displace the state across a decision boundary itself, immediately creating the possibility for failed prediction. It's also possible for $w_{synanom}$ to displace the system mildly on a particular (e.g., deeply embedded) word—not across a decision boundary, but outside of the local fg-semantic anomaly compartment. While this may produce no incorrect predictions on immediately succeeding words, its effects can be felt later once the expansive terms in the matrices of Figure 3 sufficiently magnify the displacement.

4. General Discussion

Now I turn to assessing fg-semantic and fg-syntactic anomaly in relation to natural semantic and syntactic anomaly. I will first do this by asking how this new approach fares on Challenges 1-6 above, and then return to some of the examples mentioned at the beginning of the paper that are usually taken as evidence for separate syntax and semantics.

4.1 Challenges 1-6

Regarding Challenge 1, accounting for sentences like (3) that are syntactically anomalous but understandable, the fractal grammar approach has the appealing property that it is not completely flummoxed by fg-syntactic anomaly; it continues to process and eventually recovers from the anomaly. This is in line with processing studies that indicate that humans presented with difficult garden path sentences strive to reanalyze them, though perhaps not always fully successfully (Christianson et al., 2001). Indeed, a number of proposals have been made about mechanisms of reanalysis (Ferreira and Henderson, 1998; Fodor and Inoue, 1998). Most of these are designed to address garden path sentences which temporarily mislead but then a revision of interpretation leads to a grammatical analysis. Something different is needed in the case of examples like (3), for there is no grammatical solution—for reanalysis to succeed, word addition, along with morphological modification would need to occur. And yet, while interpretation is possible in the case of examples like (3), it is not clear that the perceiver is satisfied with the sentence after understanding it, as would be expected if the system actually performed a reanalysis. So something else seems to be going on. What could this be? A natural answer is optimization—the system puts the words together as best it can, even when they do not go together perfectly. A potentially advantageous property of the fractal grammar approach is that it offers a general framework (dynamical stability) which can, under certain relatively mild conditions, be interpreted as an optimization process via the theory of Lyapunov functions. Consider a dynamical system that converges to an attractor. A Lyapunov function is a continuous function from the state space of the dynamical system to

a real number such that convergence to the attractor corresponds to gradient descent on the function. An example from physics is a function that expresses potential energy of a swinging pendulum as a function of its state (velocity and position). If there is damping, then the pendulum will eventually arrive at it lowest energy state (hanging straight down, not moving)–it optimizes its energy state by making it as low as possible. One can often identify such a function for a dynamical system that is governed by an attractor. An optimization interpretation is especially desirable in the present context because, by being explicit about the tradeoffs among different forms of solution, optimization accounts can derive heterogeneity (e.g., syntactic versus semantic badness) where other approaches are forced to stipulate it.

Regarding Challenge 2 (graded semantic anomaly) traditional approaches generally appeal to graded properties of the world. Under the traditional theory these are unrelated to gradience effects in syntax (Challenge 3). The current theory is simpler inasmuch as it is in a position to generate both kinds of gradience from the same aspect of system structure. The method of Lyapunov functions may again offer a natural avenue by supporting the definition of an energy-like quantity associated with attractor basins. On this view, higher energy states are expected to be associated with greater degrees of anomaly.

Regarding Challenge 4, the generally smaller magnitude of semantic anomaly than syntactic anomaly, the fractal grammar treatment indeed makes this claim, provided we compare cases in the same grammatical neighborhood. One may reasonably ask, Is this claim predicted by the model or is it just available to it because it has two different kinds of anomaly, one of which happens to be weaker? Put another way, is there anything that the model correctly predicts to be correlated with the kind of anomaly that is associated with an open set that contains the grammatical manifold versus the kind that is associated with the complement of this set? I am aware of one positive answer to this question: semantic anomalies in natural languages appear to lie on continua that include well-formed utterances. For example, picking up on examples (6) and (7) above, it is normal to drink water, odder to drink syrup, even more troublesome to drink motor oil, rather disturbing to

drink ball bearings, etc. Such cases illustrate how semantic anomaly, unlike syntactic anomaly is contiguous with well-formedness. In this regard, the fractal grammar account gets the story right: the milder anomaly type, fg-semantic anomaly, is the one the model treats as spatially contiguous with fully grammatical processing.

Regarding Challenge 5, "spillover effects"—the fact that the disruption associated with anomaly in human processing tends to be spread over several words following the anomalous event—stable fractal grammar models more or less predict this. Stability is characterized by a contraction of the state space around the grammatical manifold, so that, over time, the trajectories converge on the manifold. Interestingly, the degree of displacement of the system from the manifold is not monotonic in time in the model—for example, as noted above, a small disturbance in a deep center-embedding can give rise to a large disturbance several words later. There is some evidence that this might be on the right track for natural language center embeddings (King and Just, 1991; Gibson, 1998). However, there is not a clearly-established generalization in the empirical sentence processing arena about where effects will occur and how strong they will be except that they occur following anomalies and they sometimes have puzzling non-monotonicities. In fact, little work has focused on this question—further testing is needed. The current account provides predictions that could drive such investigation.

4.2 Syntax without semantics / semantics without syntax

The simple answer to how fractal grammars can handle the evidence mentioned above for syntax without semantics and semantics without syntax is that there can be layers of information structure in a dynamical system, even though the architecture is not stratified. On close inspection, the arguments for separation given in Introduction are all of a circumstantial sort: some important kind of calculation seems to involve information only of type A or only of type B. Such facts do not logically require separation of the information into two different spaces, since in each case, the unobserved type of information could simply remain unused in a particular computation even though it is available. As indicated in the main section of the paper, the

current proposal is, roughly, that the difference between syntax and semantics is a scale difference, with syntactic distinctions corresponding to bigger distances and semantic distinctions corresponding to smaller distances. This is only a rough portrayal because it glosses over the fact that the actual cutoff between small/semantic and big/syntactic is differently structured in different parts of the space. Another claim of the framework is that starting and ending in the null stack region (e.g., the middle compartment of Figure 3) as well as consistency with the labeling at every step should correspond to a judgment of grammatical well-formedness for the processed string (though not necessarily semantic well-formedness).

Although I do not yet know how to optimally write an elaborate fractal grammar for a natural language (there are many variants possible for any given symbolic system which differ in geometry and have different properties when the full range of ungrammatical strings is considered), it is nevertheless possible to see that the proposed framework offers new angles on two of the cases mentioned in Introduction.

First, regarding *likely* versus *probable*, the framework makes a prediction that could be tested empirically: even though *probable* fails to grammatically accommodate an infinitival complement via equation of lower and upper subjects, a sentence like (5d) is predicted to be judged better than a sentence like (10) and also to significantly elicit an interpretation similar to the meaning of (5c) in an unconstrained interpretation task, in contrast to (10), which is predicted to elicit no consistent interpretation.

(10) Mary is red to win the race.

The reason for these predictions is that, in a fractal grammar model of a relevant portion of English, the vector encoding of *probable* will be relatively close to the vector encoding of *likely* in virtue of their shared semantic content, so that, modulo some assumptions about the size/shape of the manifold, the sentence (5d) will cause the model to follow a trajectory that is structurally similar that of (5c), even though the model suffers an ungrammatical displacement at “probable". By contrast, the encoding of *red* is far from being compatible with the syntactic context so that sentence (10) will

produce a more severely displaced trajectory, possibly one that is not close to any complementation-structure part of the manifold after the processing of *red*.[8] The behavior that the model predicts for (5d) is related to what is called "coercion" in the linguistic literature.[9]

4.3 Why dynamical systems and why fractals?

As I noted in the introduction, the approach described in this paper diverges from classical, discrete symbolic approaches to language modeling in a way that resembles other, currently vigorously investigated divergences from the classical, and yet it also differs from these in several ways. For example, many computational linguists focus on vector spaces while few are specifically concerned with metric spaces. Also, even though it is common to introduce a norm into a vector space, which makes it a metric space, it is rare for language modelers who take this route to investigate either attractors or fractals.

Why do I think these unusual choices are warranted?

First, many of the arguments in this paper as well as many other empirical observations (e.g., Spivey, 2007) indicate that metric properties are a prominent organizational feature of the language systems of humans. Relatedly, artificial neural networks (ANNs) which learn from data (e.g., Deep Learning models) are fundamentally metric computers, for they discover structure by navigating on the basis of statistical similarity (Rumelhart et al., 1995). I suggest that the metric structure is doing much of the work for both humans and ANNs. Vector space structure is appealing for analysis purposes because linear spaces are easier to analyze—indeed, much of the headway that has been made in numerical dynamical systems theory relies on making local

[8] In making the predictions above about humans, I am banking on the assumption that alternative contexts which can make sentence (10) interpretable are so obscure that most participants will not think of them—e.g., *Mary has painted herself red for the purpose of winning the race*, *Mary's winning the race is an act of strongly showing her "red" (Native American? Republican? Communist?) side*.

[9] Villata et al. (2019) describe a dynamical coercion model which, though not a fractal grammar model, nevertheless illustrates some aspects of how linguistic coercion can work in a language processing dynamical system.

linear approximations to globally nonlinear systems. However, in naturally arising systems, including humans, I suspect linearity is not of the essence.

Second, regarding attractors, the key feature that makes attractors relevant is topological contraction: a set of higher dimension is mapped to a set of lower dimension, in many cases (including all those considered here) to a set of Lebesgue measure 0. This is a natural description of grammar: from a rich semantic world or a world rich with noise, that is to say, a world exhibiting continuum properties, one distills a discrete essence which tracks structural distinctions and nothing more. I think the interest in formalisms that bridge between discrete symbolic encodings and continuum semantic and behavioral properties, including neural network approaches (e.g., Christiansen and Chater, 1999; Elman, 1991; Devlin et al., 2019; Mikolov et al., 2013), vector-space/symbolic integration (e.g., Coecke et al., 2010; Plate, 1995; Smolensky and Legendre, 2006) and hybrids (e.g., Asher et al., 2016; Bowman, 2016; Socher, 2014; Wijnholds et al., 2020) are motivated by the insight that looking at grammar alone is an unhelpfully rarified approach—one needs to see the richness that grammar abstracts over in order to properly understand grammar itself. While vector-space/symbolic integration approaches take a helpful step in this direction, dynamical systems theory provides important additional insight by offering a principled understanding of the sources of order in symmetries (Golubitsky and Stewart, 2002).

Finally, regarding fractals, most current vector space/symbolic approaches do not mention them, but I think they might turn out to be relevant. For example, Coecke et al. (2010); Plate (1995); Smolensky and Legendre (2006) provide calculi for compositionally structured objects in vector spaces but I am not aware that any of these approaches has examined the (geo)metric structure of fully worked out infinite-state grammars within the formalisms. In all of these cases, some kind of tensor operation, involving nested instances of multiplication distributed over addition does the core work of assigning roles to entities and composing simpler entities into more complex ones. This structure is very similar to the iterated affine structure in the dynamical automata I have described above.[10] In dynamical automata it is through

[10] DA_{MR1} is not fully affine but it is piecewise affine.

this iterated interaction of multiplication and addition (in chaos theory terms, "stretching" and "folding") that fractal structures arise. A key requirement is that the computation take place on a bounded set. Boundedness seems desirable both for computational tractability and neural plausibility. Assuming that this feature is generally adopted, I suspect that fully worked out grammatical treatments in vector space/symbolic approaches will also turn out to have fractal forms.

4.4 Shortcomings

It must be acknowledged that the examples of fractal grammars given here and in other papers are all very toy examples and they give a single, arguably too narrow abstraction of natural language. For example, they do not incorporate any of the rich semantic structure that current semantic theories have worked out. The points in the fg-semantically anomalous open sets surrounding the grammatical manifold in a model like DA_{MR1} are small pieces of undifferentiated continuum. It will be important to explore richer encoding models that are closer to real natural language to see if the dynamical mechanisms will still have their beneficial properties in a richer context.

At least two additional formal features of the current model clearly need to be shifted to bring the modeling closer to human natural language processing.

One is the discrete map approximation. Much empirical work on incremental sentence processing shows that participants exhibit complex, temporally extended behavior upon processing each word. Typical measures are self-paced reading times, eye-movements in reading to points in a scene which the language is talking about, or changes in the brain's electro-magnetic field. The discrete map models employed here perform an instantaneous computation in response to each word so they are not very suitable for capturing these detailed dynamics. For this, differential equation models (among these, continuous-time recurrent neural networks) are more suitable. It is thus desirable to figure out how to implement fractal encoding in these.

At first glance, it might seem that a dynamical automata are not *compositional* in the sense of being systems that recursively combine simpler types into more complex types. Nevertheless, when it is in the middle of processing

a sentence with open dependencies, the system has a compositional encoding of the stack in the sense that its current state, interpreted as a vector, can be analyzed as the sum of a series of vectors, each at a different scale, and each of which specifies a stack symbol (see Tabor, 2011). However, compared to tensor-based encodings of semantic structure in vector space semantic models which contain full semantic information about all sentential constituents, this encoding is very impoverished.

In line with this point, a second feature that arguably needs adjusting is the empty-stack termination property that the current fractal grammars inherit from classical formal automaton design. Because it returns to the same state after every sentence, when the model has finished parsing a sentence, it has lost track of all the information expressed by the sentence. This is not realistic as a model of human sentence processing, for humans learn things from sentences. Immediately after processing them, they can often address questions or otherwise make use of the information provided by the language. Indeed, though I have advocated above for metric spaces, dynamics, and fractals over classical model-theoretic semantics and vector space semantic models, I must acknowledge that the latter two types have an advantage over current fractal grammars in that they are accumulate semantic knowledge as they progress through a sentence. A similarly informationally constructive version of fractal grammar processing is thus desirable.

4.5 Conclusion

All of these caveats are, in some sense, about the richness of the encodings. Although the caveats are nontrivial, the current approach is unusual in its ability to embrace both ideal and imperfect language processing. Robustness of this sort is a very desirable property—clearly humans have such robustness. It thus seems worth seeking a way to bring together robustness and richness.

References

Asher, N., T. Van de Cruys, and M. Abrusan. 2016. Integrating. *Computational Linguistics* 42 (4): 703–725.

Bowman, Samuel. 2016. Modeling natural language semantics in learned representations. PhD Dissertation, Stanford University.

Cho, Pyeong Whan, Matthew A. Goldrick, and Paul Smolensky. 2017. Incremental parsing in a continuous dynamical system: Sentence processing in gradient symbolic computation. *Linguistic Vanguard* 3: 76–96.

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton and Co.

Christiansen, Morten H. and Nick Chater. 1999. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science* 23: 157–205.

Christianson, Kiel, Andrew Hollingworth, John F. Halliwell, and Fernanda Ferreira. 2001. Thematic roles assigned along the garden path linger. *Cognitive Psychology* 42: 368–407.

Coecke, B., M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis* 36: 345–384.

Corballis, M. C. 2007. Recursion, language, and starlings. *Cognitive Science* 31: 697–704.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 4171–4186. Association for Computational Linguistics.

Elman, Jeffrey L. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7: 195–225.

Ferreira, Fernanda and John M. Henderson. 1998. Syntactic reanalysis, thematic processing, and sentence comprehension. In Janet D. Fodor and Fernanda Ferreira, (eds.), *Reanalysis in Sentence Processing*, pages 73–100. Dordrecht: Kluwer Academic Publishers.

Fodor, Janet Dean and Atsu Inoue. 1998. Attach anyway. In Janet D. Fodor and Fernanda Ferreira, (eds.), *Reanalysis in Sentence Processing*, pages 101–141. Dordrecht: Kluwer Academic Publishers.

Gibson, Edward. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition* 68 (1): 1–76.

Golubitsky, Martin and Ian Stewart. 2002. *The Symmetry Perspective*. Basel: Springer.

Gordon, M. J. C. 2012 [1979]. *The Denotational Description of Programming Languages: An Introduction.* New York: Springer.

Heim, Irene. 1983. File change semantics and the familiarity theory of definiteness. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow, (eds.), *Meaning, Use and Interpretation of Language*, pages 164–189. Berlin: De Gruyter.

Jackendoff, Ray. 2002. *Foundations of Language*. New York: Oxford.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic, and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers.

Keller, Frank. 2000. Gradience in grammar: Experimental and computational aspects of degrees of grammaticality. Doctoral dissertation.

Kempson, R., W. Meyer-Viol, and D. Gabbay. 2001. *Dynamic Syntax*. Oxford: Blackwell.

King, Jonathan and M.A. Just. 1991. Individual differences in syntactic processing: the role of working memory. *Journal of Memory and Language* 30: 580–602.

Kozen, Dexter C. 1997. *Automata and Complexity Theory*. New York: Springer.

Levy, Omer and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 302–308. The Association for Computational Linguistics.

Levy, Simon, Ofer Melnik, and Jordan Pollack. 2000. Infinite raam: A principled connectionist basis for grammatical competence. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society*, pages 298–303. Mahwah, NJ: Lawrence Erlbaum Associates.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th NIPS*. Lake Tahoe, Nevada, USA.

Montague, Richard. 1970. English as a formal language. In B. Visentini et al., (ed.), *Linguaggi nella e nella Tecnica*, pages 189–224. Edizioni di Comunità. Reprinted in *Formal Philosophy: Selected Papers of Richard Montague*, pp. 108–221, ed. by R. H. Thomason, New Haven: Yale University Press, 1974.

Moore, Cris. 1998. Dynamical recognizers: Real-time language recognition by analog computers. *Theoretical Computer Science* 201: 99–136.

Plate, Tony A. 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks* 6 (3): 623–641.

Rumelhart, David, Richard Durbin, Richard Golden, and Yves Chauvin. 1995. Backpropagation: The basic theory. In *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates.

Sadrzadeh, Mehrnoosh, Matthew Purver, Julian Hough, and Ruth Kempson. 2017. Exploring semantic incrementality with dynamic syntax and vector space semantics. Https://arxiv.org/abs/1811.00614.

Savitch, Walter J., (ed.). 1987. *The Formal Complexity of Natural Language*. Norwell, MA: Kluwer.

Siegelmann, H. T. and E. D. Sontag. 1994. Analog computation via neural networks. *Theoretical Computer Science* 131: 331.

Siegelmann, Hava T. 1999. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Boston: Birkhäuser.

Smolensky, Paul and Geraldine Legendre. 2006. *The Harmonic Mind: from neural computation to optimality-theoretic grammar*. Cambridge, MA: MIT Press.

Socher, Richard. 2014. Recursive deep learning for natural language processing and computer vision. PhD Dissertation, Stanford University.

Sorace, Antonella and Frank Keller. 2005. Gradience in linguistic data. *Lingua* 115 (11): 1497–1524.

Spivey, Michael. 2007. *The Continuity of Mind*. New York: Oxford University Press.

Sprouse, Jonathan, I. Caponigro, C. Greco, and C. Ceccheto. 2016. Experimental syntax and the variation of island effects in english and italian. *Natural Language and Linguistic Theory* 34: 307–344.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: The MIT Press.

Tabor, Whitney. 2000. Fractal encoding of context-free grammars in connectionist networks. *Expert Systems: The International Journal of Knowledge Engineering and Neural Networks* 17 (1): 41–56.

———. 2002. The value of symbolic computation. *Ecological Psychology* 14 (1/2): 21–52.

———. 2003. Learning exponential state growth languages by hill climbing. *IEEE Transactions on Neural Networks* 14 (2): 444–446.

———. 2009. A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cognitive Neurodynamics* 3 (4): 415–427.

———. 2011. Recursion and recursion-like structure in ensembles of neural elements. In H. Sayama, A. Minai, D. Braha, and Y. Bar-Yam, (eds.), *Unifying Themes in Complex Systems. Proceedings of the VIII International Conference on Complex Systems*, pages 1494–1508. Cambridge, MA: New England Complex Systems Institute. Http//necsi.edu/events/iccs2011/proceedings.html.

———. 2015. Fractal grammars which recover from perturbation. In *Proceedings of the NIPS Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo 2015)*. Montreal, Canada. Published on CEUR-WS: 29-Apr-2016. ISSN 1613-0073 by Creative Commons. ONLINE: http://ceur-ws.org/Vol-1583/.

Tarski, Alfred. 1933. The concept of truth in the languages of the deductive sciences (in polish). *Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznyc* 34. Expanded English translation in Tarski 1983 [1956].

———. 1983 [1956]. The concept of truth in the languages of the deductive sciences.

In John Corcoran, (ed.), *Logic, Semantics, Metamathematics: Papers from 1923 to 1938, 2nd edition*, pages 152–278. Indianapolis: Hackett Publishing Company.
Villata, Sandra. 2017. Intervention effects in sentence processing. Ph.D. Thesis, Department of Linguistics, University of Geneva.
Villata, Sandra, Luigi Rizzi, and Julie Franck. 2016. Intervention effects and relativized minimality: New experimental evidence from graded judgments. *Lingua* 179: 76–96.
Villata, Sandra, Jon Sprouse, and Whitney Tabor. 2019. Modeling ungrammaticality: A self-organizing model of islands. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society*, pages 1179–1184. Montreal, QB: Cognitive Science Society.
Villata, Sandra, Whitney Tabor, and Jon Sprouse. 2020. Gap-filling in syntactic islands: Evidence for island penetrability from the maze task. Poster presented at the 33nd annual CUNY Conference on Human Sentence Processing, Amherst, MA.
Wijnholds, Gijs, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation learning for type-driven composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324. Association for Computational Linguistics.

A. Existence of a grammatical basin

By a *grammatical basin* I mean an open set containing the grammatical manifold such that, if the system is perturbed (possibly repeatedly) in such a way that the perturbation never crosses the basin's boundary, all future transitions will be grammatical.

Thm. The grammatical manifold of dynamical automaton DA_{MR1} (Figure 3) lies in a grammatical basin.

Sketch of proof:

1. Except for those in the middle compartment, all the transition functions in Figure 3 are homeomorphisms (1-1 continuous bijections whose inverses are also continuous). Therefore, iteration up to the end of a sentence is invertible. Also, due to the topology-preserving properties of homeomorophisms, the inverse system maps partitions to partitions. Since the union of a countable infinity of partitions is a partition, the inverse future of the boundary set (the

boundary set is indicated by the dark lines in Figure 3) is a partition. I'll call this partition, IFBS, for "Inverse future of the boundary set".

2. By the *categorical future* of a point x_0 in the state space, I mean the branching sequence of symbols that can be grammatically emitted when the system is started at x_0. If two points are in the same compartment of IFBS, they have the same categorical future under within-sentence forward iteration. This is true because the only way two points can have a different categorical future is if, under grammatical iteration, they eventually arrive on different sides of a boundary. However, points in the same compartment of IFBS will never arrive at such a state during a single sentence, since IFBS is the union of all possible within-sentence histories of the boundary set.

3. The next question is whether the grammar manifold lies strictly in the interior of IFBS. IFBS can be tracked by iterating the single point, $\mathbf{p} = \begin{pmatrix} -1/2 \\ 0 \end{pmatrix}$ under the inverse language: the left boundary set is the set of all points directly above and below this point union the set of all points directly to the left of this point. The rest of the negative side of IFBS can be understood as the union of iterations of $\mathbf{p}$ under sequences of inverse b's and y's from the lefthand compartments of the partition. But these points never coincide with the boundary: the x coordinate of $\mathbf{p}$ is strictly to the left of the point $\binom{0}{0}$ associated with sentence end, and strictly to the right of the points associated with the penultimate words—$\binom{-1}{1}$ and $\binom{-1}{-1}$. Since the inverse b and y transforms are affine with positive slope, this relation is preserved under iteration. Analogous conditions ensure the persistent non-coincidence of the partition boundaries with the grammatical manifold in the right-half plane, under inverse iteration of the a and x transforms.

4. So far, these arguments imply that a point in the same compartment of IFBS as a grammatical point will have the same categorical future through sentence end. But at sentence end, since one of these two points is a grammatical point, both points have to be in the middle compartment of Figure 3. Therefore, on the next step, their two trajectories coincide. This implements contraction and makes the futures of the two trajectories identical (and grammatical) from that time on. In other words, the claims of the theorem are satisfied.