Tensor Product Representations of Subregular Formal Languages

Jonathan Rawski

Department of Linguistics and Institute for Advanced Computational Science
Stony Brook University, USA
jonathan.rawski@stonybrook.edu

Abstract

This paper provides a geometric characterization of subclasses of the regular languages. We use finite model theory to characterize objects like strings and trees as relational structures. Logical statements meeting certain criteria over these models define subregular classes of languages. The semantics of such statements can be compiled into tensor structures, using multilinear maps as function application for evaluation. This method is applied to consider two properly subregular languages over different string models.

1 Introduction

Formal language theory provides a way to explicitly tie the complexity of linguistic patterns to specific claims about memory organization and thus provides an indirect way of measuring the cognitive demands of language. The Chomsky hierarchy [Chomsky, 1959] is the standard way in formal language theory to measure language complexity. One particular class of interest is the regular languages. One of the most well-studied objects in computer science, the regular languages are characterized by regular expressions, finite-state machines, and statements in Monadic Second-Order logic, among others [Thomas, 1997]. Linguistically, the regular class has been shown to sufficiently characterize phonological and morphological patterns [Kaplan and Kay, 1994].

More recently, it has been shown that many linguistic patterns inhabit subclasses of the regular languages (see [Heinz, 2018] for an overview). These subclasses each have particular automata-theoretic, algebraic, and logical characterizations [Rogers *et al.*, 2013]. This paper considers logical characterizations of subregular language classes over various representations of structures that are defined using finite model theory.

In the last few decades, finite model theory has emerged as a powerful description language for linguistics. Model theory has been used for comparisons of particular grammatical theories in phonology and syntax [Rogers, 1998; Pullum, 2007; Graf, 2010], and for studying the nature of linguistic structures and processes themselves [Heinz, 2018]. In these formulations, linguistic structures like strings and trees are modeled using relational information which holds among the items characterizing a particular domain.

Constraints over model-theoretic structures, as well as translation between one structure and another, can be described using statements in mathematical logic. Such constraints and transformations express in an elegant way the relationship between grammars and representation. In particular, various types of logic over specific model-theoretic representations (say, strings and trees) yield particular classes of grammars [Rogers, 1996].

It is of interest to see how these models may be characterized geometrically. Geometric approaches to language and symbolic cognition in general have become increasingly popular during the last two decades. There is work dealing with conceptual spaces for sensory representations [Gardenfors, 2004], multilinear representations for compositional semantics [Blutner, 2009; Aerts, 2009], and dynamical systems for modeling language processes [Beim Graben *et al.*, 2008; Tabor, 2009].

One particularly significant contribution in this area is Smolensky's Tensor Product Representations [Smolensky, 1990]. Here, subsymbolic dynamics of neural activation patterns in a vector space description become interpreted as symbolic cognitive computations at a higher-level description by means of "filler/role" bindings via tensor products. These tensor product representations form the symbolic foundation of Harmonic Grammar and Optimality Theory, and have been successfully employed for phonological and syntactic computations. [Smolensky and Legendre, 2006].

In tensor product representations, symbolic structures are decomposed into structural roles and fillers, bound together using the tensor product. For example, strings can be decomposed into a tensor realizing string positions, each of which is bound to a tensor realizing different symbols in some alphabet. Similarly, tree structures can be represented recursively, with a tensor product representation using tree node position as the structural role, and using an entire subtree as an alphabet filler symbol. [Hale and Smolensky, 2001] use this tree representation to describe a Harmonic Grammar for context-free languages, and [Beim Graben and Gerth, 2012] use them to formalize [Stabler, 1997]'s Minimalist Grammars.

While tensor product representations form a powerful method for geometrically interpreting symbolic structures, explicitly modeling the relational structures given by model-
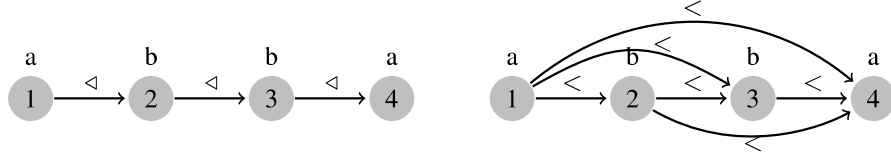
Figure 1: Visualizations of the successor (left) and precedence (right) models of *abba*.

theory, as well as the logical constraints characteristic of sub-regular languages, is an open issue. This paper provides this connection, by translating model-theoretic structures into vector spaces and describing logical grammatical constraints over them using tensors.

There has been some recent work on embedding logical calculi using tensors. [Grefenstette, 2013] introduces tensor-based predicate calculus that realizes logical operations. [Yang *et al.*, 2014] introduce a method of mining Horn clauses from relational facts represented in a vector space. [Serafini and Garcez, 2016] introduce logic tensor networks that integrate logical deductive reasoning and data-driven relational learning. [Sato, 2017] formalizes Tarskian semantics of first-order logic in vector spaces. Here we apply Sato's method for translating model-theoretic representations and first-order logic into tensors.

Finally, tensor methods and subregular grammars/automata have been used to evaluate and interpret neural networks (see [Rabanser *et al.*, 2017]). [Avcu *et al.*,] tested the generalization capacity of LSTM networks on the Strictly Local and Strictly Piecewise languages. [McCoy *et al.*, 2018] showed that recurrent neural networks (RNNs) implicitly encode tensor product representations, and [Weiss *et al.*, 2017] used regular languages to test the generalization capacity of RNNs. Explicit translation of subregular languages into tensors over various representations thus allows model-theoretic linguistics to study neural nets in a principled way.

2 Model Theoretic Representations

2.1 Elements of Finite Model Theory

Model theory, combined with logic, provides a powerful way to study and understood mathematical objects with structures [Enderton, 2001]. This paper only considers finite relational models [Libkin, 2004].

Definition 1. *A* model signature *is a tuple* $S = \langle D; R_1, R_2, \ldots, R_m \rangle$ *where the domain D is a finite set, and each R_i is a n_i-ary relation over the domain.*

In this paper, the relations are at most binary.

Definition 2. *A* model *for a set of objects Ω is a total, one-to-one function from Ω to structures whose type is given by a model signature.*

2.2 String Models

We can view strings as models. The set of all possible finite strings of symbols from a finite alphabet Σ and the set of strings of length $\leq n$ are Σ^* and $\Sigma^{\leq n}$, respectively. The unique empty string is represented with λ. The length of a string w is $|w|$, so $|\lambda| = 0$. If u and v are two strings then we denote their concatenation with uv. If w is a string and σ is the ith symbol in w then $w_i = \sigma$, so $abcd_2 = b$.

A conventional model for strings in Σ^* is given by the signature $\Gamma^\lhd = \langle D; \lhd, [R_\sigma]_{\sigma \in \Sigma} \rangle$ and the function $M^\lhd : \Sigma^* \to \Gamma^\lhd$ such that $M^\lhd(w) = \langle D^w; \lhd, [R_\sigma^w]_{\sigma \in \Sigma} \rangle$ where $D^w = \{1, \ldots, |w|\}$ is the domain, $\lhd = \{(i, i+1) \in D \times D \mid j = i+1\}$ is the successor relation which orders the elements of the domain, and $[R_\sigma^w]_{\sigma \in \Sigma}$ is a set of $|\Sigma|$ unary relations such that for each $\sigma \in \Sigma$, $R_\sigma^w = \{i \in D^w \mid w_i = \sigma\}$. We will usually omit the superscript w since it will be clear from the context.

For example, with $\Sigma = \{a, b, c\}$ and the model above for strings, we have

$$
\begin{aligned}
M^\lhd(abba) = \langle D = \{1, 2, 3, 4\}; \\
\lhd = \{(1,2), (2,3), (3,4)\}; \\
R_a = \{1, 4\}; R_b = \{2, 3\}, R_c = \emptyset \rangle \quad (1)
\end{aligned}
$$

Figure 1 illustrates $M^\lhd(abba)$ on the left.

Another conventional model is the precedence model, with the signature $\Gamma^< = \langle D; <, [R_\sigma]_{\sigma \in \Sigma} \rangle$. It differs from the successor model only in that the order relation is defined with general precedence ($<$), which is defined as $<= \{(i, j) \in D \times D \mid i < j\}$ [Büchi, 1960; McNaughton and Papert, 1971; Rogers *et al.*, 2013]. Under this model, the string *abba* has the following model.

$$
\begin{aligned}
M^<(abba) = \langle D = \{1, 2, 3, 4\}; \\
<= \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}, \\
R_a = \{1, 4\}, R_b = \{2, 3\}, R_c = \emptyset \rangle \quad (2)
\end{aligned}
$$

Figure 1 illustrates $M^<(abba)$ on the right.

Under both model signatures, each string $w \in \Sigma^*$ of length k has a unique interpretable structure. The model of string $w = \sigma_1 \sigma_2 \ldots \sigma_k$ has domain $D = \{1; 2 \ldots k\}$, and for each $\sigma \in \Sigma$, $R_\sigma = \{i \in D | w_i = \sigma\}$. The difference between $M^\lhd$ and $M^\prec$ is the ordering relation. Under the successor model $M^\lhd$, the ordering relation is $\lhd \overset{def}{=} \{(i, i+1) \in DxD\}$, while for the precedence model $M^\prec$, the ordering relation is $\prec \overset{def}{=} \{(i, j) \in D \times D | i \prec j\}$

However, structures are more general in that they correspond to any mathematical structure conforming to the model signature. As such, while a model of a string w will always be a structure, a structure will not always be a model of a string w. The *size* of a structure S, denoted $|S|$, coincides with the cardinality of its domain.

3 Logic, Languages, and Language Classes

Usually a model signature provides the vocabulary for some logical language L, which contains N constants

$\{e_1, \ldots, e_N\}$. Following notation of [Sato, 2017], a model $M = (D, I)$ is thus a pair of domain, a nonempty set D and an interpretation I that maps constants e_i to elements (entities, individuals) $I(e_i) \in D$ and k-ary predicate symbols r to k-ary relations $I(r) \subseteq D^k$

An assignment a is a mapping from variables x to an element $a(x) \in D$. It provides a way of evaluating formulas containing free variables. Syntactically terms mean variables and/or constants and atomic formulas or atoms $r(t_1, \ldots, t_k)$ are comprised of a k-ary predicate symbol r and k terms $t_1, \ldots, t_k$ some of which may be variables. Formulas F in L are inductively constructed as usual from atoms using logical connectives (negation $\neg$, conjunction $\wedge$, disjunction $\vee$) and quantifiers ($\exists, \forall$).

There are several well-known connections between logical statements and languages classes. Most famous is [Büchi, 1960]'s result that languages characterizable by finite-state machines, the regular languages, are equivalent to statements in Monadic Second-Order Logic over the precedence model for strings (and successor, since precedence is MSO-definable from successor).

Within the regular languages, many well-known subregular classes can be characterized by weakening the logic [McNaughton and Papert, 1971; Rogers *et al.*, 2013; Thomas, 1997]. An overview of these connections is shown is Figure 2
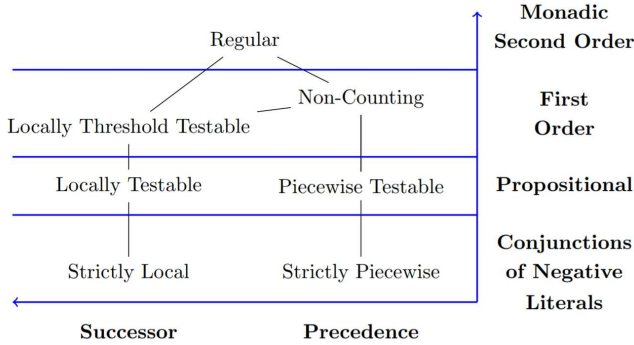

Figure 2: Subregular Hierarchy of Languages (from [Heinz, 2018])

Here we restrict ourselves to first-order logic, as it is the lightest restriction corresponding to properly subregular languages. First-order statements over particular model signatures define distinct language classes. For the successor model, [Thomas, 1997] characterizes FO($\lhd$) in terms of Local Threshold Testability, equivalence in terms of the multiplicity of k-subfactors up to some fixed finite threshold t.

Theorem 1. *([Thomas, 1997]) A set of strings is First-order definable over $\langle D; \lhd, [R_\sigma]_{\sigma \in \Sigma} \rangle$ iff it is Locally Threshold Testable.*

Correspondingly, First-order formulas over the precedence model characterize the Non-Counting or Star-Free class of languages.

Theorem 2. *([McNaughton and Papert, 1971]) Languages that are first-order definable over $\langle D; \prec, [R_\sigma]_{\sigma \in \Sigma} \rangle$ are Non-Counting.*

Non-Counting languages are those languages definable in linear temporal logic, and which have with aperiodic syntactic monoids. Since the Successor relation ($\lhd$) is first-order definable from Precedence ($\prec$), the Non-Counting class properly includes the Locally Threshold Testable class.

Further restrictions of the logic over these signatures to propositional logic or conjunctions of negative literals characterize subclasses of the LTT and Star Free languages, yielding the Local and Piecewise hierarchies. [Rogers *et al.*, 2013].

4 Tensor Representations of First-Order Logic

This section overview's [Sato, 2017]'s method for embed a model domain and signature into a vector space, using tensors to encode relational information.

Scalars are denoted with lower case letters like a. Vectors mean column vectors and we denote them by boldface lower case letters like $\mathbf{a}$ and $\mathbf{a}$'s components by a_i. $D' = \{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ is the standard basis of N-dimensional Euclidean space $\mathbb{R}^N$ where $\mathbf{e}_i = (0 \cdots, 1, \cdots, 0)^T$ is a vector that has one at the i-th position and zeros elsewhere. Such vectors are called one-hot vectors. $\mathbf{1}$ is a vector of all ones. We assume square matrices, written by boldface upper case letters like $\mathbf{A}$. $\mathbf{I}$ is an identity matrix, and $\mathbb{1}$ is a matrix of all ones. Order-p tensors $\mathcal{A} \in \mathbb{R}^{D^p}$, are also denoted by $\{a_{i_1, \ldots, i_p}\}$ $(1 \le i_1, \ldots, i_p \le N)$. $\mathcal{A}$'s component $a_{i_1, \ldots, i_p}$ is also written as $(\mathcal{A})_{i_1, \ldots, i_p}$. $(\mathbf{a} \bullet \mathbf{b}) = \mathbf{a}^T \mathbf{b}$ is the inner product of $\mathbf{a}$ and $\mathbf{b}$ whereas $\mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^T$ is their outer product. $\mathbf{1} \circ \cdots \circ \mathbf{1}$ is a k-order tensor, and $\mathbf{1} \circ \cdots \circ \mathbf{1} (\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_k}) = (\mathbf{1} \bullet \mathbf{e}_{i_1}) \cdots (\mathbf{1} \bullet \mathbf{e}_{i_k}) = 1$.

There exists an isomorphism between tensors and multilinear maps [Bourbaki, 1989], such that any curried multilinear map

$$f : V_1 \to \ldots \to V_j \to V_k$$

can be represented as a tensor $\mathcal{T}_f \in V_k \otimes V_j \otimes \ldots \otimes V_1$. This means that tensor contraction acts as function application. This isomorphism guarantees that there exists such a tensor $\mathcal{T}^f$ for every f, such for any $v_1 \in V_1, \ldots, v_j \in V_j$:

$$f \mathbf{v}_1 \ldots \mathbf{v}_j = \mathbf{v}_k = \mathcal{T}^f \times \mathbf{v}_1 \times \ldots \times \mathbf{v}_j \qquad (3)$$

Following [Sato, 2017], we first isomorphically map a model M to a model M' in $\mathbb{R}^N$. We map entities $e_i \in D$ to one-hot vectors $\mathbf{e}_i$. So D is mapped to $D' = \{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$, the basis of $\mathbb{R}^N$. We next map a k-ary relation r in M to a k-ary relation r' over D' which is computed by an order-k tensor $\mathcal{R} = \{r_{i_1, \ldots, i_k}\}$, whose truth value $[\![r(e_{i_1}, \ldots, e_{i_k})]\!]$ in M is given by

$$\begin{aligned}
&[\![r(e_{i_1}, \ldots, e_{i_k})]\!] \\
&= \mathcal{R}(\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_k}) \\
&= \mathcal{R} \times_{11} \mathbf{e}_{i_1} \times_{12} \cdots \times_{1, i_k} \mathbf{e}_{i_k} \qquad (4) \\
&= r_{i_1, \ldots, i_k} \in \{1, 0\} \; (\forall i_1, \ldots, i_k \in \{1, \ldots, N\})
\end{aligned}$$

We identity r' with $\mathcal{R}$ so that $\mathcal{R}$ encodes the M-relation r. Let M' be a model (D', I') in $\mathbb{R}^N$ such that I' interprets entities by $I'(e_i) = e_i (1 \le i \le N)$ and relations r by $I'(r) = \mathcal{R}$.

For the purposes of this paper, we restrict ourselves to binary relations and predicates. When r is a binary predicate,

the corresponding tensor $\mathcal{R}$ is a bilinear map and represented by an adjacency matrix $\mathbf{R}$ as follows:

$$[\![(e_i, e_j)]\!] = (\mathbf{e}_i \cdot \mathbf{R}\mathbf{e}_j) = \mathbf{e}_i^T \mathbf{R} \mathbf{e}_j = r_{ij} \in \{1, 0\} \quad (5)$$

Note that when $r(x, y)$ is encoded by $\mathcal{R}$ as $(\mathbf{x} \bullet \mathbf{R}\mathbf{y})$, $r(y, x)$ is encoded by $\mathbf{R}^T$, since $(\mathbf{y} \bullet \mathbf{R}\mathbf{x}) = (\mathbf{x} \bullet \mathbf{R}^T\mathbf{y})$ holds

We next inductively define the evaluation $[\![F]\!]_{I', a'}$ of a formula F in M. Let a be an assignment in M and a' the corresponding assignment in M', so $a(x) = e_i$ iff $a'(x) = \mathbf{e}_i$. For a ground atom $r(e_{i_1}, ..., e_{i_k})$, define

$$[\![r(e_{i_1}, \ldots, e_{i_k})]\!]' = \underline{\mathbf{R}}(\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_k})$$
$$(\forall i_1, \ldots, i_k \in \{1, \ldots, N\}). \quad (6)$$

where $\mathcal{R} = \{r_{i_1, \ldots, i_1}\}$ is a tensor encoding the M-relation r in M. By definition $[\![F]\!]_{I, a} = [\![F]\!]_{I, a}$ holds for any atom F. Negative literals are evaluated using $\neg\mathcal{R}$ defined as

$$[\![\neg r(e_{i_1}, \ldots, e_{i_k})]\!]' = \neg\mathcal{R}(\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_k})$$
$$\text{where } \neg\mathcal{R} \overset{def}{=} \overbrace{\mathbf{1} \circ \cdots \circ \mathbf{1}}^{k} - \mathcal{R} \quad (7)$$

$\neg\mathcal{R}$ encodes an M-relation $\neg r_1$. Negation other than negative literals, conjunction, disjunction, and quantifiers are evaluated in M' as follows.

$$[\![\neg F]\!]' = 1 - [\![F]\!]' \quad (8)$$
$$[\![F_1 \wedge \cdots \wedge F_h]\!]' = [\![F_1]\!]' \cdots [\![F_h]\!]' \quad (9)$$
$$[\![F_1 \vee \cdots \vee F_h]\!]' = \min_1([\![F_1]\!]' + \ldots + [\![F_h]\!]') \quad (10)$$
$$[\![\exists y F]\!]' = \min_1\left(\sum_{i=1}^{N} [\![F_{y \leftarrow e_i}]\!]'\right) \quad (11)$$

Here the operation $\min_1(x) = \min(x, 1) = x$ if $x < 1$, otherwise 1, as componentwise application. $F_{y \leftarrow e_i}$ means replacing every free occurrence of y in F with e_i. We treat universal quantification is treated as $\forall x F = \neg \exists x \neg F$.

5 Compiling Subregular Formulas into Tensor Algebra

[Sato, 2017] presents an algorithm for compiling any first-order formula into a tensor embedding without grounding. The algorithm works by converting a formula into prenex normal form, Each quantified statement is put in conjunctive or disjunctive normal form, depending on the quantifier, and each formula is then converted to the appropriate tensor realization.

Here we restrict ourselves to formulas with binary predicates, which as stated above may be represented as adacency matrices since their corresponding tensor is a bilinear map. Sato shows that in that in these cases, we can often "optimize" compilation by directly compiling a formula F using matrices.

5.1 Compiling a Locally Threshold Testable Formula

Here we demonstrate a formula which is properly First Order over the successor model for strings, characterizing a Locally Threshold Testable Language. With the ability to distinguish distinct occurrences of a symbol we can define a formula which is satisfied by strings containing exactly one occurrence of some symbol b. Such a system is seen in phonological stress patterns in the world's languages, which often mandate exactly one primary stress in a word. We do this by asserting that there is some position in which b occurs $\exists x (R_b(x))$, and that there are no other positions in which b occurs $\wedge (\forall y)[R_b(y) \rightarrow (x = y)]]$. The conjunction of these two gives the FO formula

$$F_{\text{one-}B} = (\exists x \forall y)[R_b(x) \wedge [R_b(y) \rightarrow (x = y)]] \quad (12)$$

Converting this into prenex normal form we get

$$\exists x \forall y (R_B(x) \wedge [\neg R_B(y) \vee (x = y)] \quad (13)$$

Compiling this formula into tensor notation is rather straightforward.

$$\mathcal{T}_{\text{one-}B} =$$
$$min_1\left(\sum_{i=1}^{N} 1 - min_1\left(\sum_{j=1}^{N} \mathcal{R}^b \mathbf{e}_i \bullet [(1 - \mathcal{R}^b \mathbf{e}^j) + \right.\right.$$
$$\left.\left. + (\mathbf{e}_i \bullet \mathbf{e}_j)]\right)\right) \quad (14)$$

Intuitively, this formula checks whether two domain elements are the same via the inner product, and if both domain elements have the property of being a b, then the formula evaluates to 1. If either formula is not a b, or if two different domain emelents are a b, the formula evaluates to 0. We can apply this to to the successor model for *abba* in (1) by defining each of the relational tensors in the formula over the domainand relations in the model. Doing so, it is easy to see that the formula evaluates to 0 (false) for domain elements 2 and 3, which are distinct and each have the property of being a b.

5.2 Compiling a Non-Counting Formula

Next we demonstrate a formula which is properly First Order over the precedence model for strings, characterizing a Non-Counting language. We motivate this formula using a phonological pattern from Latin, in which in certain cases an l cannot follow another l unless an r intervenes, no matter the distance between them [Jensen, 1974; Heinz, 2010]. This can be seen in the $-alis$ adjectival suffix which appears as $-aris$ if the word it attaches to already contains an l, except in cases where there is an intervening r, in which it appears again as -alis.

The blocking effects in this non-local alternating pattern requires the use of quantifiers, and is properly Non-Counting. We can represent it with the following first-order formula:

$$F_{diss} = \forall x \forall y [R_l(x) \wedge R_l(y) \wedge R_{\prec}(x, y) \rightarrow$$
$$\rightarrow \exists z [R_r(z) \wedge R_{\prec}(x, z) \wedge R_{\prec}(z, y)] \quad (15)$$

Converting this into prenex normal form we get

$$\exists x \exists y \exists z \neg [R_l(x) \wedge R_l(y) \wedge R_\prec(x,y)] \vee$$
$$\vee [R_r(z) \wedge R_\prec(x,z) \wedge R_\prec(z,y)] \quad (16)$$

The compilation of this formula is again quite straightforward:

$$\mathcal{T}_{diss} = min_1\Big(\sum_{i=1}^{N} min_1\Big(\sum_{j=1}^{N} min_1\Big(\sum_{k=1}^{N}$$
$$1 - \Big[(\mathcal{R}^l \mathbf{e}_i) \bullet (\mathcal{R}^l \mathbf{e}_j) \bullet (\mathbf{e}_i \mathcal{R}^\prec \mathbf{e}_j)\Big] +$$
$$+ \Big[(\mathcal{R}^z \mathbf{e}_k) \bullet (\mathbf{e}_i \mathcal{R}^\prec \mathbf{e}_k) \bullet (\mathbf{e}_k \mathcal{R}^\prec \mathbf{e}_j)\Big]\Big)\Big)\Big) \quad (17)$$

Intuitively, this formula tests whether, for any two domain elements labeled l and precede each other, there is another element labeled r which comes between them. The use of the precedence relation here shows that this can happen anywhere in the word, and can thus handle the Latin dissimilation patterns above.

6 Extension to Tree Structures

The model-theory framework also allows describing tree structures. [Rogers, 2003] describes a model-theoretic characterization of trees of arbitrary dimensionality. In his framework, we specify the domain D as a Gorn tree domain [Gorn, 1967]. This is a hereditarily prefix closed set D of node addresses, that is to say, for every $d \in D$ with $d := \alpha i$, it holds that $\alpha \in D$, and for every $d \in D$ with $d := \alpha i \neq \alpha 0$, then $\alpha(i-1) \in D$. In this view, a string may be called a one-dimensional or unary-branching tree, since it has one axis along which its nodes are ordered. In a standard tree, on the other hand, the set of nodes is ordered as above by two relations, "dominance" and "immediate left-of". Suppose s is the mother of two nodes t and u in some standard tree, and also assume that t precedes u. Then we might say that s dominates the string tu.
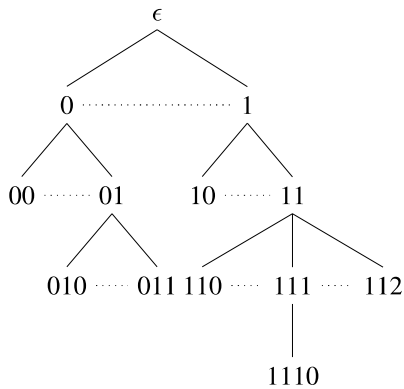

Figure 3: 2-dimensional tree model. Dominance and precedence relations shown with solid/dashed and dotted lines, respectively

While a Gorn tree domain as written encodes these dominance and precedence relations implicitly, we may explicitly write them out model-theoretically so that a signature for a Σ-labeled 2-d tree T is $\Gamma^\lhd = \langle D; \lhd, \prec, [R_\sigma]_{\sigma \in \Sigma} \rangle$ where $\lhd$ is the immediate dominance relation and $\prec$ is the "immediate left-of" relation. Model signatures that include transitive closure relations of each of these have also been studied.

We can thus generalize strings as 1-dimensional tree models, and standard trees as two-dimensional trees, which relate nodes to one-dimensional trees by immediate dominance. A three-dimensional tree relates nodes to two-dimensional, i.e. standard trees. In general, a d-dimensional tree is a set of nodes ordered by d dominance relations such that the n-th dominance relation relates nodes to $(n-1)$-dimensional trees (for $d = 1$, single nodes are zero-dimensional trees). Importantly, we may compile these into tensors without the recursive role embeddings in Smolensky's formulation.

7 Conclusion

This paper provided a method for geometrically characterizing subregular languages in vector spaces. Model-theoretic descriptions of relational structures were embedded in Euclidean vector spaces, and statements in first-order logic over these structures were compiled into tensor formulas. Semantic evaluation was given via tensor contraction over tensors implementing a specific model. This method can easily be extended to consider other relational structures, and to other logics. Another application is to consider logical translations between model signatures, which define mappings between structures [Courcelle, 1994], another area relevant for linguistics. The analytical power given by multilinear algebra, combined with the representational flexibility given by finite model theory and mathematical logic, provides a powerful combination for analyzing the nature of linguistic structures and cognition, and for exploring the relationship of languages and computation more generally.

References

[Aerts, 2009] Diederik Aerts. Quantum structure in cognition. *Journal of Mathematical Psychology*, 53(5):314–348, 2009.

[Avcu et al.,] Enes Avcu, Chihiro Shibata, and Jeffrey Heinz. Subregular complexity and deep learning. pages 20–33.

[Beim Graben and Gerth, 2012] Peter Beim Graben and Sabrina Gerth. Geometric representations for minimalist grammars. *Journal of Logic, Language and Information*, 21(4):393–432, 2012.

[Beim Graben et al., 2008] Peter Beim Graben, Dimitris Pinotsis, Douglas Saddy, and Roland Potthast. Language processing with dynamic fields. *Cognitive Neurodynamics*, 2(2):79–88, 2008.

[Blutner, 2009] Reinhard Blutner. Concepts and bounded rationality: An application of niestegge's approach to conditional quantum probabilities. In *AIP Conference Proceedings*, volume 1101, pages 302–310. AIP, 2009.

[Bourbaki, 1989] Nicolas Bourbaki. *Commutative Algebra: Chapters 1-7*. Springer-Verlag, 1989.

[Büchi, 1960] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

[Chomsky, 1959] Noam Chomsky. On certain formal properties of grammars. *Information and Control*, 2:137–167, 1959.

[Courcelle, 1994] Bruno Courcelle. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75, 1994.

[Enderton, 2001] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2nd edition, 2001.

[Gardenfors, 2004] Peter Gardenfors. Conceptual spaces as a framework for knowledge representation. *Mind and Matter*, 2(2):9–27, 2004.

[Gorn, 1967] Saul Gorn. Explicit definitions and linguistic dominoes. In J. Hart and S. Takasu, editors, *Systems and Computer Science*, pages 77–115, Toronto, 1967. University of Toronto Press.

[Graf, 2010] Thomas Graf. Logics of phonological reasoning. Master's thesis, University of California, Los Angeles, 2010.

[Grefenstette, 2013] Edward Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, 2013.

[Hale and Smolensky, 2001] John Hale and Paul Smolensky. A parser for harmonic context-free grammars. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 427–432, 2001.

[Heinz, 2010] Jeffrey Heinz. Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661, 2010.

[Heinz, 2018] Jeffrey Heinz. The computational nature of phonological generalizations. In Larry Hyman and Frans Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton, 2018.

[Jensen, 1974] John Jensen. Variables in phonology. *Language*, 50(4):675–686, 1974.

[Kaplan and Kay, 1994] Ronald M Kaplan and Martin Kay. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378, 1994.

[Libkin, 2004] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[McCoy et al., 2018] R Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. Rnns implicitly implement tensor product representations. *arXiv preprint arXiv:1812.08718*, 2018.

[McNaughton and Papert, 1971] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, 1971.

[Pullum, 2007] Geoffrey K. Pullum. The evolution of model-theoretic frameworks in linguistics. In James Rogers and Stephan Kepser, editors, *Model-Theoretic Syntax at 10*, pages 1–10, Dublin, Ireland, 2007.

[Rabanser et al., 2017] Stephan Rabanser, Oleksandr Shchur, and Stephan Günnemann. Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*, 2017.

[Rogers et al., 2013] James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. Cognitive and sub-regular complexity. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer, 2013.

[Rogers, 1996] James Rogers. Strict lt 2: Regular:: local: recognizable. In *International Conference on Logical Aspects of Computational Linguistics*, pages 366–385. Springer, 1996.

[Rogers, 1998] James Rogers. *A descriptive approach to language-theoretic complexity*. CSLI Publications Stanford, CA, 1998.

[Rogers, 2003] James Rogers. Syntactic structures as multi-dimensional trees. *Research on Language and Computation*, 1(3-4):265–305, 2003.

[Sato, 2017] Taisuke Sato. Embedding tarskian semantics in vector spaces. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Serafini and Garcez, 2016] Luciano Serafini and Artur S d'Avila Garcez. Learning and reasoning with logic tensor networks. In *Conference of the Italian Association for Artificial Intelligence*, pages 334–348. Springer, 2016.

[Smolensky and Legendre, 2006] Paul Smolensky and Géraldine Legendre. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, volume Volume I: Cognitive Architecture. MIT Press, 2006.

[Smolensky, 1990] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216, 1990.

[Stabler, 1997] Edward P. Stabler. Derivational minimalism. In Christian Retoré, editor, *Logical aspects of computational linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–195, Berlin, 1997. Springer.

[Tabor, 2009] Whitney Tabor. A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cognitive neurodynamics*, 3(4):415–427, 2009.

[Thomas, 1997] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, chapter 7. Springer, 1997.

[Weiss et al., 2017] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. *arXiv preprint arXiv:1711.09576*, 2017.

[Yang et al., 2014] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.